514-3IDD/0195

MISSION OPERATIONS AND DATA SYSTEMS DIRECTORATE

Landsat 7 Processing System (LPS) Interface Definitions Document (IDD)

October 1996

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland

Landsat 7 Processing System (LPS)
Interface Definitions Document (IDD)

October 1996

Rachel Shea 11/7/96

Rachel Shea Date
RDCS, Lead
Landsat 7 Processing System
Code 514
Goddard Space Flight Center

Michael R. Reid 11/6/96

Michael Reid Date
IDPS, Lead
Landsat 7 Processing System
Code 514
Goddard Space Flight Center

Ives Wu 11-6-96

Ives Wu Date
RDPS, Lead
Landsat 7 Processing System
Code 514
Goddard Space Flight Center

Hyun Soon Kim Date
MACS / Database, Lead
Landsat 7 Processing System
Code 514
Goddard Space Flight Center

11/11/96

Lakshmi Gupta Date
MFPS, Lead
Landsat 7 Processing System
Code 514
Goddard Space Flight Center

Shujun Huang 11-6-96

Shu-Jun Huang Date
LDTS, Lead
Landsat 7 Processing System
Code 514
Goddard Space Flight Center

Minnie Wong 11-6-96

Minnie Wong Date
PCDS, Lead
Landsat 7 Processing System
Code 514
Goddard Space Flight Center

11-6-96

Viet Trinh Date
Global Library, Lead
Landsat 7 Processing System
Code 514
Goddard Space Flight Center

Landsat 7 Processing System (LPS)
Interface Definitions Document (IDD)

October 1996

Concurred by:

11/7/96

Daniel Ferry, Technical Lead — Date
Landsat 7 Processing System, Code 514

Concurred by:

Nathaniel E. Daniel — 11-7-96

Nathaniel Daniel, Program Manager — Date
Landsat 7 Processing System, Code 514

Approved by:

Robert Schweiss — 11/7/96

Robert Schweiss, System Engineer — Date
Landsat 7 Processing System, Code 514

Approved by:

11/7/96

Jeffrey Hosler, Software Manager — Date
Landsat 7 Processing System, Code 514

Approved by:

Joy Henegar — 11/7/96

Joy Henegar, Project Manager — Date
Landsat 7 Processing System, Code 514

Abstract

This Interface Definitions Document (IDD) presents the functional, performance, operational, and design requirements for the interfaces between the Landsat 7 Processing System (LPS) subsystems.

This document provides a current understanding of the definition of the interfaces between the LPS subsystems. This interface definitions document will be baselined by the LPS during the LPS detailed design activities.

Keywords: *Interface Definitions Document (IDD), Landsat 7 Processing System (LPS)*

Preface

This IDD is controlled by the Mission Operations and Systems Development Division (MOSDD) Configuration Control Board (CCB) and may be updated by Document Change Notice (DCN) or revision.

The interface definitions were prepared by:

Landsat 7 Processing System Project
Code 514.0
Goddard Space Flight Center
Greenbelt, MD 20771

GSFC

C. Brambora
L. Cisney
M. Dowdy
J. Henegar
J. Hosler
E. Lee
R. Schweiss

CNMOS

T. Aslam
D. Crehan
E. Criscuolo
N. Daniel
D. Ferry
S. Gottsagen
L. Gupta
M. Huang
X. Jiang
H. Kim
Y. Kim
S. Lee
C. Liu
R. Mazzola
D. Nguyen
Y. O
M. Reid
R. Shea
E. Staples
R. Tingley
V. Trinh
H. Vinulan
M. Wallace
S. Whisonat
M. Wong
I. Wu

Table of Contents

Section 1. Introduction

1.1 Purpose 1-1

1.2 Scope 1-1

1.3 Organization 1-1

1.4 Applicable Documents 1-1

1.4.1 Specification Documents 1-2

1.4.2 Reference Documents 1-3

Section 2. MACS and All LPS Subsystems

2.1 Process Invocation 2-1

2.1.1 Description 2-1

2.1.2 Data Transfer 2-1

2.1.3 RDCS 2-2

2.1.4 RDPS 2-3

2.1.5 MFPS 2-3

2.1.6 PCDS 2-3

2.1.7 IDPS 2-4

2.1.8 LDTS 2-4

2.2 Processing Parameters and Thresholds 2-5

2.2.1 Processing Parameters 2-5

2.2.1.1 Description 2-5

2.2.1.2 RDPS 2-5

2.2.1.3 MFPS 2-7

2.2.1.4 PCDS 2-15

2.2.1.5 IDPS 2-18

2.2.1.6 LDTS 2-20

2.2.2 Error Reporting Thresholds 2-23

2.2.2.1 Description...2-23

2.2.2.2 RDPS...2-23

2.2.2.3 MFPS...2-25

2.2.2.4 PCDS...2-27

2.3 LPS Journal Messages...2-30

2.3.1 Description...2-30

2.3.2 Format...2-30

2.3.3 IPC Mechanism..2-30

2.3.4 Data Transfer...2-30

2.3.5 Frequency..2-31

2.3.6 Sizing..2-31

2.4 Graphical User Interface..2-32

2.4.1 Description...2-32

Section 3. Metadata and Quality and Accounting

3.1 Metadata Information...3-1

3.1.1 Description...3-1

3.1.2 RDCS 3-1

3.1.3 MFPS 3-4

3.1.4 PCDS..3-9

3.1.5 IDPS...3-20

3.2 Quality and Accounting Information.....................................3-26

3.2.1 Description...3-26

3.2.2 RDCS 3-26

3.2.3 RDPS..3-28

3.2.4 MFPS 3-32

Section 4. RDCS and RDPS

4.1 Raw Data Capture File 4-1
4.1.1 Description 4-1
4.1.2 Format 4-1
4.1.3 IPC Mechanism 4-1
4.1.4 Data Transfer 4-1
4.1.5 Frequency 4-1
4.1.6 Sizing 4-1

Section 5. RDPS and MFPS

5.1 Annotated CADUs 5-1
5.1.1 Description 5-1
5.1.2 Format 5-1
5.1.3 IPC Mechanism 5-3
5.1.4 Data Transfer 5-3
5.1.5 Frequency 5-4
5.1.6 Sizing 5-4

Section 6. MFPS and PCDS

6.1 PCD Information 6-1
6.1.1 Description 6-1
6.1.2 Format 6-1
6.1.3 IPC Mechanism 6-3
6.1.4 Data Transfer 6-3
6.1.5 Frequency 6-3
6.1.6 Sizing 6-4

6.2 Subinterval Information 6-4
6.2.1 Description 6-4
6.2.2 Format 6-5
6.2.3 IPC Mechanism 6-5

6.2.4 Data Transfer....................6-5
6.2.5 Frequency....................6-6
6.2.6 Sizing....................6-7

Section 7. MFPS and IDPS

7.1 Aligned Bands....................7-1
7.1.1 Description....................7-1
7.1.2 Format....................7-1
7.1.3 IPC Mechanism....................7-4
7.1.4 Data Transfer....................7-5
7.1.5 Frequency....................7-5
7.1.6 Sizing....................7-6
7.2 Subinterval Information....................7-6
7.2.1 Description....................7-6
7.2.2 Format....................7-6
7.2.3 IPC Mechanism....................7-7
7.2.4 Data Transfer....................7-7
7.2.5 Frequency....................7-7
7.2.6 Sizing....................7-7

Section 8. PCDS and IDPS

8.1 Scene Information 8-1
8.1.1 Description....................8-1
8.1.2 Format....................8-1
8.1.3 IPC Mechanism....................8-3
8.1.4 Data Transfer....................8-3
8.1.5 Frequency....................8-3
8.1.6 Sizing....................8-4

Section 9. Product File Information

9.1 Product Filename Information....................9-1

9.1.1 Description....................9-1

9.1.2 RDCS....................9-1

9.1.3 MFPS....................9-3

9.1.4 MACS....................9-4

9.2 LDTS Product File Information....................9-6

9.2.1 Description....................9-6

9.2.2 Format....................9-6

9.2.3 IPC Mechanism....................9-7

9.2.4 Data Transfer....................9-7

9.2.5 Frequency....................9-9

9.2.6 Sizing....................9-9

Appendix A. Oracle 7 Data Sizing

Appendix B. Acronym List

List of Figures

Figure 5-1. Annotated CADU 5-1

Figure 6-1. PCD Information..6-1

Figure 7-1. Aligned Bands (Format 1) 7-1

Figure 7-2. Aligned Bands (Format 2) 7-2

Figure 8-1. Scene Information..8-1

List of Tables

Table 2-1. RDCS Process Arguments....................2-2

Table 2-2. RDPS Process Arguments....................2-3

Table 2-3. MFPS Process Arguments....................2-3

Table 2-4. PCDS Process Arguments....................2-3

Table 2-5. IDPS Process Arguments....................2-4

Table 2-6. LDTS Process Arguments 2-4

Table 2-7. RDPS Processing Parameters Key / Search Information....................2-5

Table 2-8. RDPS Processing Parameters 2-5

Table 2-9. RDPS Processing Parameters Sizing....................2-7

Table 2-10. MFPS Processing Parameters....................2-7

Table 2-11. MFPS Processing Parameters Sizing....................2-15

Table 2-12. PCDS Processing Parameters 2-15

Table 2-13. PCDS Processing Parameters Sizing....................2-17

Table 2-14. IDPS Processing Parameters 2-18

Table 2-15. IDPS Processing Parameters Sizing....................2-19

Table 2-16. LDTS Processing Parameters....................2-20

Table 2-17. LDTS Processing Parameters Sizing 2-22

Table 2-18. RDPS Error Reporting Thresholds 2-23

Table 2-19. RDPS Error Reporting Thresholds Sizing....................2-24

Table 2-20. MFPS Error Reporting Thresholds....................2-25

Table 2-21. MFPS Error Reporting Thresholds Sizing....................2-27

Table 2-22. PCDS Error Reporting Thresholds 2-27

Table 2-23. PCDS Error Reporting Thresholds Sizing..........2-29

Table 2-24. LPS Message Priorities..........2-30

Table 3-1. RDCS Metadata Key / Search Information..........3-1

Table 3-2. RDCS Metadata Information..........3-2

Table 3-3. RDCS Metadata Information Sizing..........3-4

Table 3-4. MFPS Metadata Key / Search Information..........3-4

Table 3-5. MFPS Metadata Information..........3-4

Table 3-6. MFPS Metadata Information Sizing..........3-9

Table 3-7. PCDS Metadata Key / Search Information..........3-9

Table 3-8. PCDS Metadata Information..........3-10

Table 3-9. PCDS Metadata Information Sizing..........3-19

Table 3-10. IDPS Metadata Key / Search Information..........3-20

Table 3-11. IDPS Metadata Information..........3-20

Table 3-12. IDPS Metadata Information Sizing..........3-25

Table 3-13. RDCS Quality and Accounting Key / Search Information..........3-26

Table 3-14. RDCS Quality and Accounting Information..........3-27

Table 3-15. RDCS Quality and Accounting Information Sizing..........3-28

Table 3-16. RDPS Quality and Accounting Key / Search Information..........3-28

Table 3-17. RDPS Quality and Accounting Information..........3-29

Table 3-18. RDPS Quality and Accounting Information Sizing..........3-32

Table 3-19. MFPS Quality and Accounting Key / Search Information..........3-32

Table 3-20. MFPS Quality and Accounting Information..........3-32

Table 3-21. MFPS Quality and Accounting Information Sizing..........3-37

Table 5-1. Annotated CADU Structure (lpsAnnotatedCADU)..........5-1

Table 5-2. Annotated CADU Interface Sizing..5-4

Table 6-1. PCD Information Structure (lpsPCDInfoStruct) 6-2

Table 6-2. VCDU Information Structure (lpsVcduInfoStruct) 6-2

Table 6-3. PCD Information Interface Sizing...6-4

Table 6-4. Subinterval Key / Search Information...6-5

Table 6-5. Subinterval Information...6-5

Table 6-6. Subinterval Information Interface Sizing...6-7

Table 7-1. Major Frame Structure (lps_MajorFrame_TYPE)......................................7-2

Table 7-2. Band 6 Scan Data Structure (lps_Band6_Scan_Data_TYPE).........................7-3

Table 7-3. Format 1 Scan Data Structure (lps_Format1_Scan_Data_TYPE)....................7-3

Table 7-4. Format 2 Scan Data Structure (lps_Format2_Scan_Data_TYPE)....................7-4

Table 7-5. Time Structure (lps_TimeStruct)..7-4

Table 7-6. Aligned Bands Interface Sizing...7-6

Table 8-1. Scene Information Structure (lps_Scene_Info_TYPE)................................8-2

Table 8-2. Geographical Coordinate Structure (lps_GeoCoord_TYPE) 8-2

Table 8-3. Scene Information Interface Sizing 8-4

Table 9-1. RDCS Product Filename Key / Search Information 9-1

Table 9-2. RDCS Product Filename Information 9-1

Table 9-3. RDCS Product Filename Information Sizing...9-3

Table 9-4. MFPS Product Filename Key / Search Information 9-3

Table 9-5. MFPS Product Filename Information 9-3

Table 9-6. MFPS Product Filename Information Sizing...9-4

Table 9-7. MACS Product Filename Key / Search Information...................................9-4

Table 9-8. MACS Product Filename Information...9-5

Table 9-9. MACS Product Filename Information Sizing....9-6

Table 9-10. LDTS Product File Key / Search Information....9-6

Table 9-11. LDTS Product File Information....9-7

Table 9-12. Filename Format....9-8

Table 9-13. File Identification Type....9-8

Table 9-14. LDTS Product File Information Interface Sizing....9-9

Table A-1. Oracle 7 Integer Data Type Sizing....A-1

Table A-2. Oracle 7 Real Data Type Sizing....A-1

Table A-3. Oracle 7 Miscellaneous Data Type Sizing....A-1

Section 1. Introduction

1.1 Purpose

This Interface Definitions Document (IDD) presents the interface requirements between the Landsat 7 Processing System (LPS) subsystems located at the Earth Resources Observation System (EROS) Data Center (EDC).

This document is an incorporated part of the LPS Detailed Design Specification (DDS). The purpose of this document is to provide further detail regarding the requirements for the interfaces described in the LPS DDS.

1.2 Scope

This document provides the functional, performance, operational, and design requirements for the LPS subsystem interfaces. This document is intended for all parties requiring such information, including system engineers and system designers responsible for implementing the interfaces, and the system maintenance personnel responsible for maintaining the interfaces.

1.3 Organization

This document is organized into nine sections. Section 1 is the introduction to this document. The remaining sections are organized into four general categories.

The first category includes all Management and Control Subsystem (MACS) interfaces related to the Graphical User Interface (GUI), processing control, and message reporting. This includes the GUI items provided by the LPS subsystems to the MACS, the process communication information, processing parameters, and error reporting thresholds provided by the MACS to the LPS subsystems, and the LPS Journal messages provided by the LPS subsystems to the MACS. The behavior of these interfaces is virtually identical from subsystem to subsystem, so the interfaces are grouped together in section 2.

The second category includes all of the metadata and quality and accounting information provided by the LPS subsystems to MACS. The behavior of these interfaces is virtually identical from subsystem to subsystem, so the interfaces are grouped together in section 3.

The third category includes the major subsystem interfaces. These interfaces are unique in content, but generally involve telemetry data. This category is comprised of sections 4-8, with each section describing a subsystem-to-subsystem grouping of interfaces.

The fourth category includes all output product file information transferred from the Raw Data Capture Subsystem (RDCS), the Major Frame Processing Subsystem (MFPS), and the Management and Control Subsystem (MACS) to the subsystems that produce the output product files, and from the subsystems that produce the output product files to the LPS Data Transfer Subsystem (LDTS). These interfaces are described in section 9.

1.4 Applicable Documents

The following documents contain additional details regarding the LPS, the Landsat 7 System and Project, and external systems.

1.4.1 Specification Documents

These documents, provide the basis for developing the LPS subsystem interface definitions presented in this document.

1. National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC)/Mission Operations and Data System Directorate (MO&DSD), *Landsat 7 Processing System (LPS) Functional and Performance Specification (F&PS)*, 560-8FPS/0194, Revision 1 w/ DCN 2, July 31, 1996
2. NASA GSFC/MO&DSD, *Landsat 7 Processing System (LPS) Operations Concept*, Revision 2, 560-3OCD/0194, April 15, 1996
3. NASA GSFC/MO&DSD, *Landsat 7 Processing System (LPS) System Design Specification (SDS)*, 560-8SDS/0194, May 26, 1995
4. NASA GSFC/MO&DSD, *Landsat 7 Processing System (LPS) Software Requirements Specification (SRS)*, 560-8SWR/0195, April 28, 1995
5. NASA GSFC/MO&DSD, *Landsat 7 Processing System (LPS) Detailed Design Specification* , 514-4DDS/0195, November 1995
6. Consultative Committee for Space Data Systems (CCSDS), *Recommendation for Space Data System Standards; Advanced Orbiting Systems (AOS), Networks and Data Links: Architectural Specification*, Blue Book, CCSDS 701.0-B-2, Issue 2, November 1992
7. Lockheed Martin Astro Space (LMAS), *Landsat 7 System Data Format Control Book (DFCB), Volume 4 , Revision C—Wideband Data*, 23007702, April 4, 1996
8. NASA GSFC/MO&DSD, *Landsat 7 Processing System (LPS) Output Files Data Format Control Book*, 510-3FCD/0195,June 14, 1996
9. NASA GSFC, *Interface Control Document (ICD) Between the Landsat 7 Ground Station (LGS) and the Landsat 7 Processing System (LPS)*, Revision 1, 560-1ICD/0794, Oct 17, 1996
10. NASA GSFC, *Interface Control Document (ICD) between the ESDIS Core System (ECS) and the Landsat 7 System*, 209-CD-013-004, August 1996 [Note: includes LPS-EDC DAAC interface requirements]
11. NASA GSFC, *Memo of Understanding (MOU) Between the Landsat 7 Processing System (LPS) and the Mission Operations Center (MOC)*, May 1995
12. NASA GSFC, *Interface Control Document (ICD) Between the Image Assessment System (IAS) and the Landsat 7 Processing System (LPS)*, Revision 1, 514-1ICD/0195, July 29, 1996
13. Computer Sciences Corporation (CSC), *SEAS System Development Methodology*, July 1989

1.4.2 Reference Documents

These documents are used as sources of additional and background information, as required, for developing the LPS interface definitions.

1. NASA GSFC/MO&DSD, *Landsat 7 Processing System (LPS) User's Guide for Release 1 DRAFT* , 514-3SUG/0195, January 1997
2. NASA GSFC/MO&DSD, *Systems Management Policy*, MDOD-8YMP/0485, July 1986
3. NASA GSFC/MO&DSD, *Landsat 7 Detailed Mission Requirements*, May 15, 1995
4. NASA GSFC/MO&DSD, *Landsat 7 Ground Station (LGS) Functional and Performance Specification (F&PS)*, Review, 531-FPS-GN/Landsat 7, January 1996
5. NASA GSFC/MO&DSD, *Mission Operations Concept Document for the Landsat 7 Ground System*, June 5, 1995
6. Martin Marietta Astro Space, *Landsat 7 Image Assessment System (IAS) Operations Concept*, Landsat 7 Library No. 5527, September, 1994
7. National Aeronautics and Space Administration (NASA), *Landsat 7 Level 1 Requirements*, Draft Issue, August 8, 1994
8. United States Geological Survey (USGS)/National Oceanic and Atmospheric Administration (NOAA), *Index to Landsat Worldwide Reference System (WRS) Landsats 1, 2, 3, and 4*, 1982
9. Martin Marietta Astro Space, *Landsat 7 System Program Coordinates System Standard*, proposed update draft, 23007610A, August 1994
10. NASA GSFC/MO&DSD, *Landsat 7 Processing System (LPS) ProjectManagement Plan*, Revision 1, May 1995
11. NASA GSFC/MO&DSD, *Landsat 7 System and Operations Concept* (Pre-CCB Baseline version), 430-11-06--003-0, October 1994

Section 2. MACS and All LPS Subsystems

This section describes all interfaces related to the Graphical User Interface (GUI), processing control, and message reporting between the MACS and all other LPS subsystems.

2.1 Process Invocation

This subsection describes all information transferred between the MACS and all other LPS subsystems via process invocation.

2.1.1 Description

The LPS subsystem processes are designed to be executed from the UNIX command line with command line arguments. The MACS provides the user a Graphical User Interface (GUI) to these processes. MACS, therefore, will provide necessary information to the subsystem processes in the same manner that they would be executed from the command line.

2.1.2 Data Transfer

The MACS primary communication mechanism with all other LPS subsystems is through process invocation.

MACS spawns processes passing information via command line arguments. The arguments are prefixed with an argument identifier followed by an optional value, e.g. *ss_Process -a value1 -b -c value3 ...* Arguments without values denote a boolean flag. All arguments will be passed as an ASCII character string to the spawned processes.

The MACS will monitor the spawned, or child, processes, and will retrieve subsystem status information via return values captured through the use of the UNIX signals. Specifically, the signals will be obtained by capturing the signal SIGCHLD with a MACS signal handler upon the completion of the spawned process. The process status provided by the UNIX Kernel will be passed to the global routine lps_ProcessChildStatus to obtain the actual return value from the child process. The ordering of the Application Process Interface (API) can be obtained from the unit lps_ProcessChildStatus.

The child processes will use explicit calls to exit() or return() to provide their return values to the MACS.

The details of the subsystem processes are available in the following subsections. For the "Format/Type" fields of the following tables, additional types have been added:

- TS - Date/time with one second resolution, format YYDDDHHMMSS where
  - YY- Last two digits of year, e.g. "92" for 1992
  - DDD- Day of year from "001" to "366"

  - HH - Hour of day from "00" to "23"
  - MM - Minute of hour from "00" to "59"
  - SS - Second of minute from "00" to "59"
- AI(n) - ASCII string representing an integer with a range from 1 to n digits, leading 0's are permitted

2.1.3 RDCS

Table 2-1. RDCS Process Arguments

Process Name	Argument(s)					Return Value
	Description	Option	Default	Format/ Type	Range	
rdc_Delete Files	Raw Data Filename Including Path Delete Raw Data File Only	-f /path/name -o	(Required) Delete both raw data and accnting file	Char [255] N/A	N/A N/A	LPS_SUCCESS LPS_FAILURE
rdc_GenReport	Contact Sequence ID	-c ID	(Required)	AI(10)	1-ULONG_MAX	LPS_SUCCESS LPS_FAILURE
rdc	Isolate Processor Suspend L0R Stop Time Start Time (Log Only) Capture Source ID	-i -s -e time -b time -l ID	Not isolated Not suspended Max contact NULL (Required)	N/A N/A TS TS Char [2]	N/A N/A N/A N/A 1I,1Q, 2I,2Q, 3I,3Q, TB, T1,T2,T3,T4,T5 (see table 9-2)	LPS_SUCCESS LPS_FAILURE
rdc_Restage	Media Device Name	-d device	/dev/rmt8	Char [255]	N/A	LPS_SUCCESS LPS_FAILURE
rdc_Save	Raw Data Filename Including Path Media Device Name	-f /path/name -d device	(Required) /dev/rmt8	Char [255] Char [255]	N/A N/A	LPS_SUCCESS LPS_FAILURE
rdc_StopSave Restage	Save or Restage Mode	-m	Save mode	N/A	N/A	LPS_SUCCESS LPS_FAILURE
rdc_Terminate	N/A	N/A	N/A	N/A	N/A	0 - Success 1 - Failure
rdc_TestData	Isolate Processor Suspend L0R Stop Time Start Time (Log Only) Capture Source ID	-i -s -e time -b time -l ID	Not isolated Not suspended Max contact NULL (Required)	N/A N/A TS TS Char [2]	N/A N/A N/A N/A 1I,1Q, 2I,2Q, 3I,3Q, TB, T1,T2,T3,T4,T5 (see table 9-2)	LPS_SUCCESS LPS_FAILURE
rdc_UpdAcct	N/A	N/A	N/A	N/A	N/A	LPS_SUCCESS LPS_FAILURE

2.1.4 RDPS

Table 2-2. RDPS Process Arguments

Process Name	Argument(s)					Return Value
	Description	Option	Default	Format/ Type	Range	
rdp	Raw Data Filename Including Path	-f /path/name	(Required)	Char [255]	N/A	LPS_SUCCESS LPS_FAILURE
	Always BCH after CRC	-B	No BCH after good CRC	N/A	N/A	
	Contact Sequence ID	-c ID	(Required)	AI(10)	1-ULONG_MAX	
	Version Number	-v ID	(Required)	AI(10)	0-9	

2.1.5 MFPS

Table 2-3. MFPS Process Arguments

Process Name	Argument(s)					Return Value
	Description	Option	Default	Format/ Type	Range	
mfp	Raw Data Filename Including Path	-f /path/name	(Required)	Char [255]	N/A	LPS_SUCCESS LPS_FAILURE
	Contact Sequence ID	-c ID	(Required)	AI(10)	1-ULONG_MAX	
	Version Number	-v ID	(Required)	AI(10)	0-9	

2.1.6 PCDS

Table 2-4. PCDS Process Arguments

Process Name	Argument(s)					Return Value
	Description	Option	Default	Format/ Type	Range	
pcd	Contact Sequence ID	-c ID	(Required)	AI(10)	1-ULONG_MAX	LPS_SUCCESS
	Version Number	-v ID	(Required)	AI(10)	0-9	LPS_FAILURE

2.1.7 IDPS

Table 2-5. IDPS Process Arguments

Process Name	Argument(s)					Return Value
	Description	Option	Default	Format/ Type	Range	
idp	Contact Sequence ID Version Number	-c ID -v ID	(Required) (Required)	AI(10) AI(10)	1-ULONG_MAX 0-9	LPS_SUCCESS LPS_FAILURE LPS_ERROR

2.1.8 LDTS

Table 2-6. LDTS Process Arguments

Process Name	Argument(s)					Return Value
	Description	Option	Default	Format/ Type	Range	
deletefile	Contact Sequence ID Version Number Override Retention Delete Transferred Files Only	-c ID -v ID -R -o	(Required) (Required) No override Delete all files	AI(10) AI(10) N/A N/A	1-ULONG_MAX 0-9 N/A N/A	LPS_SUCCESS LPS_FAILURE
genfts	Stop Time Start Time	-e time -b time	Midnight of current day Current time	TS TS	N/A N/A	LPS_SUCCESS LPS_FAILURE
rcvddn	N/A	N/A	N/A	N/A	N/A	LPS_SUCCESS LPS_FAILURE
resenddan	N/A	N/A	N/A	N/A	N/A	LPS_SUCCESS LPS_FAILURE
retainfile	Contact Sequence ID Version Number	-c ID -v ID	(Required) (Required)	AI(10) AI(10)	1-ULONG_MAX 0-9	LPS_SUCCESS LPS_FAILURE
senddan	Contact Sequence ID Version Number Resend DAN	-c ID -v ID -r	(Required) (Required) Send DAN for first time	AI(10) AI(10) N/A	N/A N/A	LPS_SUCCESS LPS_FAILURE
stopddn	N/A	N/A	N/A	N/A	N/A	LPS_SUCCESS LPS_ERROR

2.2 Processing Parameters and Thresholds

This subsection describes all processing parameters and error reporting thresholds transferred between the MACS and the LPS subsystems.

2.2.1 Processing Parameters

2.2.1.1 Description

The Processing Parameters interface contains all processing parameters transferred from the MACS to the RDPS, MFPS, PCDS, IDPS, and LDTS subsystems.

2.2.1.2 RDPS

2.2.1.2.1 Format

The RDPS processing parameters key / search information are detailed in table 2-7.

Table 2-7. RDPS Processing Parameters Key / Search Information

Table	Attribute	Type	Range	Description
Valid_CCSDS_Parms	CCSDS_Parms_Id	Number(10)	1-ULONG_MAX (Not NULL)	CCSDS parameters sequence ID

The RDPS processing parameters are detailed in table 2-8.

Table 2-8. RDPS Processing Parameters

Table	Attribute	Type	Range	Description
Valid_CCSDS_Parms	Insertion_Time	Date	(Not NULL)	Record insertion time
Valid_CCSDS_Parms	CADU_Search_Tol	Number(1)	1-3 (Not NULL)	Search tolerance parameter
Valid_CCSDS_Parms	CADU_Check_Tol	Number(1)	0-3 (Not NULL)	Check tolerance parameter
Valid_CCSDS_Parms	CADU_Flywheel_Tol	Number(1)	0-3 (Not NULL)	Flywheel tolerance parameter
Valid_CCSDS_Parms	CADU_Sync_Mark_ Check_Error_Tol	Number(1)	0-3 (Not NULL)	Check error tolerance parameter
Valid_CCSDS_Parms	CADU_Sync_Lock_ Error_Tol	Number(1)	0-3 (Not NULL)	Lock error tolerance parameter
Valid_CCSDS_Parms	CADU_Bit_Slip_Corr_ Extent	Number(1)	0-3 (Not NULL)	Bit slip correction extent parameter

2.2.1.2.2 IPC Mechanism

The IPC mechanism for the RDPS Processing Parameters interface is the LPS database. The database table name consisting of the processing parameters is called "Valid_CCSDS_Parms". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

2.2.1.2.3 Data Transfer

The MACS and RDPS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Before the MACS initiates Level 0R processing, it may modify, on operator command, any or all of the RDPS processing parameters. If changes to the RDPS processing parameters are required, the MACS must create a new database table record in Valid_CCSDS_Parms to record the parameter changes. Within the new record, the following may be modified.

Set:

- Valid_CCSDS_Parms(CCSDS_Parms_Id) = a unique CCSDS parameter sequence ID

Update (Modify the CADU search tolerance):

- Valid_CCSDS_Parms(CADU_Search_Tol)

Update (Modify the CADU check tolerance):

- Valid_CCSDS_Parms(CADU_Check_Tol)

Update (Modify the flywheel tolerance):

- Valid_CCSDS_Parms(CADU_Flywheel_Tol)

Update (Modify the CADU synchronization check error tolerance):

- Valid_CCSDS_Parms(CADU_Sync_Mark_Check_Error_Tol)

Update (Modify the CADU synchronization lock error tolerance):

- Valid_CCSDS_Parms(CADU_Sync_Lock_Error_Tol)

Update (Modify the CADU bit slip correction extent):

- Valid_CCSDS_Parms(CADU_Bit_Slip_Corr_Extent)

The MACS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database, and prior to initiating Level 0R processing. Upon committing the changes to the database, the MACS must record the record insertion time by performing the following.

Update (Modify the record insertion time):

- Valid_CCSDS_Parms(Insertion_Time) = the current system time

The RDPS will extract the processing parameters for the contact period, prior to beginning Level 0R processing of the contact period. RDPS must provide the following information.

Obtain (Extract the CCSDS parameters):

- Valid_CCSDS_Parms(CADU_Search_Tol, CADU_Check_Tol, CADU_Flywheel_Tol, CADU_Sync_Mark_Check_Error_Tol, CADU_Sync_Lock_Error_Tol, CADU_Bit_Slip_Corr_Extent)

Where:

- Valid_CCSDS_Parms.Insertion_Time IN

Obtain (Locate the latest CCSDS parameters record):

  - MAX(Valid_CCSDS_Parms.Insertion_Time)

2.2.1.2.4 Frequency

The MACS is expected to make available to the RDPS the contact period processing parameters prior to initiating Level 0R processing for the contact period. Modifications to the processing parameters cannot occur during Level 0R processing of any contact period on the associated LPS string.

2.2.1.2.5 Sizing

Table 2-9 specifies the sizing for the RDPS Processing Parameters interface. The numbers are based on the following:

- Oracle 7 relational Database Management System (DBMS) data type sizing
- A quality and accounting retention time of 30 days
- A nominal contact period reprocessing count of **TBD** per 30 days

Table 2-9. RDPS Processing Parameters Sizing

Interface Element	Number	Size (Bytes)
Valid_CCSDS_Parms Record	1	25
Valid_CCSDS_Parms Record (30 days)	**TBD**	**TBD**

2.2.1.3 MFPS

2.2.1.3.1 Format

The MFPS processing parameters are detailed in table 2-10.

Table 2-10. MFPS Processing Parameters

Table	Attribute	Type	Range	Description
Valid_MFP_Parms	Sensor_Align_Fwd_W_B1_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_B1_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_B1_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_B1_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_B2_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment

Table 2-10. MFPS Processing Parameters (Cont.)

Table	Attribute	Type	Range	Description
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B2_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B2_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B2_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B3_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B3_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B3_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B3_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B4_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B4_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B4_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B4_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B5_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B5_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B5_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B5_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B6_15	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B6_15	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B6_26	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B6_26	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B6_37	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B6_37	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B6_48	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B6_48	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B7_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B7_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B7_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_ B7_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_W_ B8_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment

Table 2-10. MFPS Processing Parameters (Cont.)

Table	Attribute	Type	Range	Description
Valid_MFP_Parms	Sensor_Align_Fwd_W_B8_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_B8_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Fwd_E_B8_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B1_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B1_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_B1_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_B1_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B2_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B2_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_B2_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_B2_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B3_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B3_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_B3_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_B3_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B4_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B4_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_B4_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_B4_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B5_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B5_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_B5_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_B5_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B6_15	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_B6_15	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B6_26	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_B6_26	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_B6_37	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment

Table 2-10. MFPS Processing Parameters (Cont.)

Table	Attribute	Type	Range	Description
Valid_MFP_Parms	Sensor_Align_Rev_E_ B6_37	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_ B6_48	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_ B6_48	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_ B7_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_ B7_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_ B7_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_ B7_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_ B8_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_W_ B8_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_ B8_Even	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Sensor_Align_Rev_E_ B8_Odd	Number(5)	0-99999 (Not NULL)	Value used to perform odd/even pixel alignment
Valid_MFP_Parms	Fill_Value	Char(1) (**TBR**)	(Not NULL)	Value to fill missing ETM+ major frame data
Valid_MFP_Parms	Sub_Intv_Delta	Number(9,2)	0.00 - 9999999.99 (Not NULL)	The minimum time in milliseconds used to determine if a new subinterval is to be declared
Valid_MFP_Parms	Mjf_Data_Period	Number(9,2)	0.00 - 9999999.99 (Not NULL)	The nominal data rate threshold of ETM+ major frames per millisecond
Valid_MFP_Parms	Max_Alignment_Value	Number(5)	0-99999 (Not NULL)	Maximum allowable pixel alignment value
Valid_MFP_Parms	Time_Range_Tol	Number(9,2)	0.00 - 9999999.99 (Not NULL)	The time range tolerance in 1/16th msec
Valid_MFP_Parms	Part_Mnf_Tol	Number(8)	0-99999999 (Not NULL)	The minimum number of data word groups required before a partial ETM+ minor frame is considered readable
Valid_MFP_Parms	Maj_Vote_Tol	Number(8)	0-99999999 (Not NULL)	The value indicating the minimum number of identical bits that are required for the data word group majority vote

2.2.1.3.2 IPC Mechanism

The IPC mechanism for the MFPS Processing Parameters interface is the LPS database. The database table name consisting of the processing parameters is called "Valid_MFP_Parms". The Valid_MFP_Parms table is a single record table. The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

2.2.1.3.3 Data Transfer

The MACS and MFPS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Before the MACS initiates Level 0R processing, it may modify, on operator command, any or all of the MFPS processing parameters. Within the record, the following may be modified.

Update (Modify the pixel alignment parameter for Band 1, forward, west, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B1_Even)

Update (Modify the pixel alignment parameter for Band 1, forward, west, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B1_Odd)

Update (Modify the pixel alignment parameter for Band 1, forward, east, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B1_Even)

Update (Modify the pixel alignment parameter for Band 1, forward, east, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B1_Odd)

Update (Modify the pixel alignment parameter for Band 2, forward, west, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B2_Even)

Update (Modify the pixel alignment parameter for Band 2, forward, west, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B2_Odd)

Update (Modify the pixel alignment parameter for Band 2, forward, east, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B2_Even)

Update (Modify the pixel alignment parameter for Band 2, forward, east, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B2_Odd)

Update (Modify the pixel alignment parameter for Band 3, forward, west, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B3_Even)

Update (Modify the pixel alignment parameter for Band 3, forward, west, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B3_Odd)

Update (Modify the pixel alignment parameter for Band 3, forward, east, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B3_Even)

Update (Modify the pixel alignment parameter for Band 3, forward, east, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B3_Odd)

Update (Modify the pixel alignment parameter for Band 4, forward, west, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B4_Even)

Update (Modify the pixel alignment parameter for Band 4, forward, west, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B4_Odd)

Update (Modify the pixel alignment parameter for Band 4, forward, east, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B4_Even)

Update (Modify the pixel alignment parameter for Band 4, forward, east, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B4_Odd)

Update (Modify the pixel alignment parameter for Band 5, forward, west, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B5_Even)

Update (Modify the pixel alignment parameter for Band 5, forward, west, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B5_Odd)

Update (Modify the pixel alignment parameter for Band 5, forward, east, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B5_Even)

Update (Modify the pixel alignment parameter for Band 5, forward, east, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B5_Odd)

Update (Modify the pixel alignment parameter for Band 6, forward, west, 15 detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B6_15)

Update (Modify the pixel alignment parameter for Band 6, forward, west, 15 detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B6_15)

Update (Modify the pixel alignment parameter for Band 6, forward, east, 26 detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B6_26)

Update (Modify the pixel alignment parameter for Band 6, forward, east, 26 detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B6_26)

Update (Modify the pixel alignment parameter for Band 6, forward, west, 37 detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B6_37)

Update (Modify the pixel alignment parameter for Band 6, forward, west, 37 detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B6_37)

Update (Modify the pixel alignment parameter for Band 6, forward, east, 48 detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B6_48)

Update (Modify the pixel alignment parameter for Band 6, forward, east, 48 detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B6_48)

Update (Modify the pixel alignment parameter for Band 7, forward, west, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B7_Even)

Update (Modify the pixel alignment parameter for Band 7, forward, west, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B7_Odd)

Update (Modify the pixel alignment parameter for Band 7, forward, east, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B7_Even)

Update (Modify the pixel alignment parameter for Band 7, forward, east, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B7_Odd)

Update (Modify the pixel alignment parameter for Band 8, forward, west, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B8_Even)

Update (Modify the pixel alignment parameter for Band 8, forward, west, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B8_Odd)

Update (Modify the pixel alignment parameter for Band 8, forward, east, even detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B8_Even)

Update (Modify the pixel alignment parameter for Band 8, forward, east, odd detector):

- Valid_MFP_Parms(Sensor_Align_Fwd_E_B8_Odd)

[NOTE: The above update statements also apply for the reverse pixel alignment values.]

Update (Modify the fill value):

- Valid_MFP_Parms(Fill_Value)

Update (Modify the subinterval delta parameter):

- Valid_MFP_Parms(Sub_Intv_Delta)

Update (Modify the ETM+ major frame data period):

- Valid_MFP_Parms(Mjf_Data_Period)

Update (Modify the maximum pixel alignment value):

- Valid_MFP_Parms(Max_Alignment_Value)

Update (Modify the time range tolerance):

- Valid_MFP_Parms(Time_Range_Tol)

Update (Modify the minimum number of data word groups required for an ETM+ minor frame):

- Valid_MFP_Parms(Part_Mnf_Tol)

Update (Modify the majority vote tolerance):

- Valid_MFP_Parms(Maj_Vote_Tol)

The MACS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database, and prior to initiating Level 0R processing.

The MFPS will extract the processing parameters for the contact period, prior to beginning Level 0R processing of the contact period. MFPS must provide the following information.

Obtain (Extract the MFP parameters):

- Valid_MFP_Parms(Sensor_Align_Fwd_W_B1_Even, Sensor_Align_Fwd_W_B1_Odd, Sensor_Align_Fwd_E_B1_Even, Sensor_Align_Fwd_E_B1_Odd, Sensor_Align_Fwd_W_B2_Even, Sensor_Align_Fwd_W_B2_Odd, Sensor_Align_Fwd_E_B2_Even, Sensor_Align_Fwd_E_B2_Odd, Sensor_Align_Fwd_W_B3_Even, Sensor_Align_Fwd_W_B3_Odd, Sensor_Align_Fwd_E_B3_Even, Sensor_Align_Fwd_E_B3_Odd, Sensor_Align_Fwd_W_B4_Even, Sensor_Align_Fwd_W_B4_Odd, Sensor_Align_Fwd_E_B4_Even, Sensor_Align_Fwd_E_B4_Odd, Sensor_Align_Fwd_W_B5_Even, Sensor_Align_Fwd_W_B5_Odd, Sensor_Align_Fwd_E_B5_Even, Sensor_Align_Fwd_E_B5_Odd, Sensor_Align_Fwd_W_B6_15, Sensor_Align_Fwd_E_B6_15, Sensor_Align_Fwd_W_B6_26, Sensor_Align_Fwd_E_B6_26, Sensor_Align_Fwd_W_B6_37, Sensor_Align_Fwd_E_B6_37, Sensor_Align_Fwd_W_B6_48, Sensor_Align_Fwd_E_B6_48, Sensor_Align_Fwd_W_B7_Even, Sensor_Align_Fwd_W_B7_Odd, Sensor_Align_Fwd_E_B7_Even, Sensor_Align_Fwd_E_B7_Odd, Sensor_Align_Fwd_W_B8_Even, Sensor_Align_Fwd_W_B8_Odd, Sensor_Align_Fwd_E_B8_Even, Sensor_Align_Fwd_E_B8_Odd, Sensor_Align_Rev_W_B1_Even, Sensor_Align_Rev_W_B1_Odd, Sensor_Align_Rev_E_B1_Even, Sensor_Align_Rev_E_B1_Odd, Sensor_Align_Rev_W_B2_Even, Sensor_Align_Rev_W_B2_Odd, Sensor_Align_Rev_E_B2_Even, Sensor_Align_Rev_E_B2_Odd, Sensor_Align_Rev_W_B3_Even, Sensor_Align_Rev_W_B3_Odd, Sensor_Align_Rev_E_B3_Even, Sensor_Align_Rev_E_B3_Odd, Sensor_Align_Rev_W_B4_Even, Sensor_Align_Rev_W_B4_Odd, Sensor_Align_Rev_E_B4_Even, Sensor_Align_Rev_E_B4_Odd, Sensor_Align_Rev_W_B5_Even, Sensor_Align_Rev_W_B5_Odd, Sensor_Align_Rev_E_B5_Even, Sensor_Align_Rev_E_B5_Odd, Sensor_Align_Rev_W_B6_15, Sensor_Align_Rev_E_B6_15, Sensor_Align_Rev_W_B6_26, Sensor_Align_Rev_E_B6_26, Sensor_Align_Rev_W_B6_37, Sensor_Align_Rev_E_B6_37, Sensor_Align_Rev_W_B6_48, Sensor_Align_Rev_E_B6_48, Sensor_Align_Rev_W_B7_Even, Sensor_Align_Rev_W_B7_Odd, Sensor_Align_Rev_E_B7_Even, Sensor_Align_Rev_E_B7_Odd, Sensor_Align_Rev_W_B8_Even, Sensor_Align_Rev_W_B8_Odd, Sensor_Align_Rev_E_B8_Even, Sensor_Align_Rev_E_B8_Odd, Fill_Value, Sub_Intv_Delta, Mjf_Data_Period, Max_Alignment_Value, Time_Range_Tol, Part_Mnf_Tol, Maj_Vote_Tol)

2.2.1.3.4 Frequency

The MACS is expected to make available to the MFPS the contact period processing parameters prior to initiating Level 0R processing for the contact period. Modifications to the processing parameters cannot occur during Level 0R processing of any contact period on the associated LPS string.

2.2.1.3.5 Sizing

Table 2-11 specifies the sizing for the MFPS Processing Parameters interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing

Table 2-11. MFPS Processing Parameters Sizing

Interface Element	Number	Size (Bytes)
Valid_MFP_Parms Record	1	**321**

2.2.1.4 PCDS

2.2.1.4.1 Format

The PCDS processing parameters are detailed in table 2-12.

Table 2-12. PCDS Processing Parameters

Table	Attribute	Type	Range	Description
Valid_PCD_Parms	PCD_Frame_Fill	Number(8)	0-99999999 (Not NULL)	Value used to fill missing PCD data
Valid_Scene_Parms	ETM_Body_Trans_ Matrix_11	Number(3,2)	-1.00 - 1.00 (Not NULL)	Transformation matrix cell 1,1 from the ETM+ line-of-sight at the center of the scan, to the spacecraft body
Valid_Scene_Parms	ETM_Body_Trans_ Matrix_12	Number(3,2)	-1.00 - 1.00 (Not NULL)	Transformation matrix cell 1,2
Valid_Scene_Parms	ETM_Body_Trans_ Matrix_13	Number(3,2)	-1.00 - 1.00 (Not NULL)	Transformation matrix cell 1,3
Valid_Scene_Parms	ETM_Body_Trans_ Matrix_21	Number(3,2)	-1.00 - 1.00 (Not NULL)	Transformation matrix cell 2,1
Valid_Scene_Parms	ETM_Body_Trans_ Matrix_22	Number(3,2)	-1.00 - 1.00 (Not NULL)	Transformation matrix cell 2,2
Valid_Scene_Parms	ETM_Body_Trans_ Matrix_23	Number(3,2)	-1.00 - 1.00 (Not NULL)	Transformation matrix cell 2,3
Valid_Scene_Parms	ETM_Body_Trans_ Matrix_31	Number(3,2)	-1.00 - 1.00 (Not NULL)	Transformation matrix cell 3,1
Valid_Scene_Parms	ETM_Body_Trans_ Matrix_32	Number(3,2)	-1.00 - 1.00 (Not NULL)	Transformation matrix cell 3,2
Valid_Scene_Parms	ETM_Body_Trans_ Matrix_33	Number(3,2)	-1.00 - 1.00 (Not NULL)	Transformation matrix cell 3,3
Valid_Scene_Parms	Mission_Start_Time	Date	(Not NULL)	Start time of the Landsat 7 mission (sec)
Valid_Scene_Parms	Time_Per_Orbit	Number(5,2)	(Not NULL)	Amount of time required for the Landsat 7 spacecraft to make one complete orbit (unit **TBD**)
Valid_Scene_Parms	Semi_Major_Axis	Number(10,3)	0.00-**TBD** (Not NULL)	Distance from Apogee or Perigee to the center of the orbit ellipse

Table 2-12. PCDS Processing Parameters (Cont.)

Table	Attribute	Type	Range	Description
Valid_Scene_Parms	Semi_Minor_Axis	Number(10,3)	0.00-**TBD** (Not NULL)	Polar axis radius
Valid_Scene_Parms	ETM_Plus_LOS_X	Number(14,7)	0.00-**TBD** (Not NULL)	X-coordinate of the line-of-sight vector
Valid_Scene_Parms	ETM_Plus_LOS_Y	Number(14,7)	0.00-**TBD** (Not NULL)	Y-coordinate of the line-of-sight vector
Valid_Scene_Parms	ETM_Plus_LOS_Z	Number(14,7)	0.00-**TBD** (Not NULL)	Z-coordinate of the line-of-sight vector

2.2.1.4.2 IPC Mechanism

The IPC mechanism for the PCDS Processing Parameters interface is the LPS database. The database table names consisting of the processing parameters are called "Valid_PCD_Parms" and "Valid_Scene_Parms". The Valid_PCD_Parms table and the Valid_Scene_Parms table are single record tables. The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

2.2.1.4.3 Data Transfer

The MACS and PCDS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Before the MACS initiates Level 0R processing, it may modify, on operator command, any or all of the PCDS processing parameters. Within the record, the following may be modified.

Update (Modify the PCD data fill value):

- Valid_PCD_Parms(PCD_Frame_Fill)

Update (Modify the ETM+ to spacecraft body transformation matrix):

- Valid_Scene_Parms(ETM_Body_Trans_Matrix_11, ETM_Body_Trans_Matrix_12 , ETM_Body_Trans_Matrix_13, ETM_Body_Trans_Matrix_21, ETM_Body_Trans_Matrix_22, ETM_Body_Trans_Matrix_23, ETM_Body_Trans_Matrix_31, ETM_Body_Trans_Matrix_32, ETM_Body_Trans_Matrix_33)

Update (Modify the Landsat 7 mission start time):

- Valid_Scene_Parms(Mission_Start_Time)

Update (Modify the Landsat 7 orbit time):

- Valid_Scene_Parms(Time_Per_Orbit)

Update (Modify the distance from Apogee or Perigee to the center of the orbit ellipse):

- Valid_Scene_Parms(Semi_Major_Axis)

Update (Modify the polar axis radius):

- Valid_Scene_Parms(Semi_Minor_Axis)

Update (Modify the line-of-sight XYZ coordinate):

- Valid_Scene_Parms(ETM_Plus_LOS_X, ETM_Plus_LOS_Y, ETM_Plus_LOS_Z)

The MACS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database, and prior to initiating Level 0R processing.

The PCDS will extract the processing parameters for the contact period, prior to beginning Level 0R processing of the contact period. PCDS must provide the following information.

Obtain (Extract the PCD parameters):

- Valid_PCD_Parms(PCD_Frame_Fill)

Obtain (Extract the scene parameters):

- Valid_Scene_Parms(ETM_Body_Trans_Matrix_11, ETM_Body_Trans_Matrix_12, ETM_Body_Trans_Matrix_13, ETM_Body_Trans_Matrix_21, ETM_Body_Trans_Matrix_22, ETM_Body_Trans_Matrix_23, ETM_Body_Trans_Matrix_31, ETM_Body_Trans_Matrix_32, ETM_Body_Trans_Matrix_33, Mission_Start_Time, Time_Per_Orbit, Semi_Major_Axis, Semi_Minor_Axis, ETM_Plus_LOS_X, ETM_Plus_LOS_Y, ETM_Plus_LOS_Z)

2.2.1.4.4 Frequency

The MACS is expected to make available to the PCDS the contact period processing parameters prior to initiating Level 0R processing for the contact period. Modifications to the processing parameters cannot occur during Level 0R processing of any contact period on the associated LPS string.

2.2.1.4.5 Sizing

Table 2-13 specifies the sizing for the PCDS Processing Parameters interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing

Table 2-13. PCDS Processing Parameters Sizing

Interface Element	Number	Size (Bytes)
Valid_PCD_Parms Record	1	**5**
Valid_Scene_Parms Record	1	**77**

2.2.1.5 IDPS

2.2.1.5.1 Format

The IDPS processing parameters are detailed in table 2-14.

Table 2-14. IDPS Processing Parameters

Table	Attribute	Type	Range	Description
Valid_Band_Parms	Multi1	Number(1)	1-8 (Not NULL)	Specifies the first of three bands to process for Browse and ACCA
Valid_Band_Parms	Multi2	Number(1)	1-8 (Not NULL)	Specifies the second of three bands to process for Browse and ACCA
Valid_Band_Parms	Multi3	Number(1)	1-8 (Not NULL)	Specifies the third of three bands to process for Browse and ACCA
Valid_Band_Parms	Subs	Number(3)	4,8,16,32,48 (Not NULL) (**TBR**)	Subsampling reduction ratio
Valid_Band_Parms	Wave	Number(3)	(**TBD**) (Not NULL)	Wavelet iterations
Valid_Band_Parms	CCA_Method	Varchar2(256)	(Not NULL)	Describes the cloud cover assessment method
Valid_Band_Parms	CCA_Ratio	Number(3)	4,8,16,32,48 (Not NULL)	Cloud cover assessment reduction ratio

2.2.1.5.2 IPC Mechanism

The IPC mechanism for the IDPS Processing Parameters interface is the LPS database. The database table name consisting of the processing parameters is called "Valid_Band_Parms". The Valid_Band_Parms table is a single record table. The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

2.2.1.5.3 Data Transfer

The MACS and IDPS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Before the MACS initiates Level 0R processing, it may modify, on operator command, any or all of the IDPS processing parameters. Within the record, the following may be modified.

Update (Modify the Browse and ACCA bands to process):

- Valid_Band_Parms(Multi1, Multi2, Multi3)

Update (Modify the subsampling reduction ratio):

- Valid_Band_Parms(Sub)

Update (Modify the wavelet iterations):

- Valid_Band_Parms(Wave)

Update (Modify the cloud cover assessment method):

- Valid_Band_Parms(CCA_Method)

Update (Modify the cloud cover assessment reduction ratio):

- Valid_Band_Parms(CCA_Ratio)

The MACS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database, and prior to initiating Level 0R processing.

The IDPS will extract the processing parameters for the contact period, prior to beginning Level 0R processing of the contact period. IDPS must provide the following information.

Obtain (Extract the IDP parameters):

- Valid_Band_Parms(Multi1, Multi2, Multi3, Subs, Wave, CCA_Method, CCA_Ratio)

2.2.1.5.4 Frequency

The MACS is expected to make available to the IDPS the contact period processing parameters prior to initiating Level 0R processing for the contact period. Modifications to the processing parameters cannot occur during Level 0R processing of any contact period on the associated LPS string.

2.2.1.5.5 Sizing

Table 2-15 specifies the sizing for the IDPS Processing Parameters interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing

Table 2-15. IDPS Processing Parameters Sizing

Interface Element	Number	Size (Bytes)
Valid_Band_Parms Record	1	**271**

2.2.1.6 LDTS

2.2.1.6.1 Format

The LDTS processing parameters are detailed in table 2-16.

Table 2-16. LDTS Processing Parameters

Table	Attribute	Type	Range	Description
Valid_LDT_Parms	Num_Auth_Request	Number(4)	0-9999 (Not NULL)	Number of times to attempt to send an authentication request
Valid_LDT_Parms	Timeout_Auth_Request	Number(4)	0-9999 (Not NULL)	Specifies the amount of time to wait for acknowledgment of an authentication request before timing out (sec)
Valid_LDT_Parms	Num_Send_DAN_Attempt	Number(4)	0-9999 (Not NULL)	Number of times to attempt to send a DAN
Valid_LDT_Parms	Timeout_Send_DAN_ Attempt	Number(4)	0-9999 (Not NULL)	Specifies the amount of time to wait while attempting to send a DAN before timing out (sec)
Valid_LDT_Parms	Timeout_Receive_DAA	Number(4)	0-9999 (Not NULL)	Specifies the amount of time to wait for the receipt of a DAA before timing out (sec)
Valid_LDT_Parms	Timeout_Receive_DDN	Number(4)	0-9999 (Not NULL)	Specifies the amount of time to wait for the receipt of a DDN before timing out (sec)
Valid_LDT_Parms	Num_Send_DDA_Attempt	Number(4)	0-9999 (Not NULL)	Number of times to attempt to send a DDA
Valid_LDT_Parms	Timeout_Send_DDA	Number(4)	0-9999 (Not NULL)	Specifies the amount of time to wait while attempting to send a DDA before timing out (sec)
Valid_LDT_Parms	Read_Sleep_Second	Number(10)	1-ULONG_MAX (Not NULL)	Sleep time between socket reads (sec)

2.2.1.6.2 IPC Mechanism

The IPC mechanism for the LDTS Processing Parameters interface is the LPS database. The database table name consisting of the processing parameters is called "Valid_LDT_Parms". The Valid_LDT_Parms table is a single record table. The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

2.2.1.6.3 Data Transfer

The MACS and LDTS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Before the MACS initiates Level 0R processing, it may modify, on operator command, any or all of the LDTS processing parameters. Within the record, the following may be modified.

Update (Modify the number of times to attempt to send an authentication request):

- Valid_LDT_Parms(Num_Auth_Request)

Update (Modify the time to wait for acknowledgment of an authentication request before timing out (sec)):

- Valid_LDT_Parms(Timeout_Auth_Request)

Update (Modify the Number of times to attempt to send a DAN):

- Valid_LDT_Parms(Num_Send_DAN_Attempt)

Update (Modify the amount of time to wait while attempting to send a DAN before timing out (sec)):

- Valid_LDT_Parms(Timeout_Send_DAN_Attempt)

Update (Modify the amount of time to wait for the receipt of a DAA before timing out (sec)):

- Valid_LDT_Parms(Timeout_Receive_DAA)

Update (Modify the amount of time to wait for the receipt of a DDN before timing out (sec)):

- Valid_LDT_Parms(Timeout_Receive_DDN)

Update (Modify the number of times to attempt to send a DDA):

- Valid_LDT_Parms(Num_Send_DDA_Attempt)

Update (Modify the amount of time to wait while attempting to send a DDA before timing out (sec)):

- Valid_LDT_Parms(Timeout_Send_DDA)

Update (Modify the sleep time between socket reads (sec)):

- Valid_LDT_Parms(Read_Sleep_Second)

The MACS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database, and prior to initiating Level 0R processing.

The LDTS will extract the processing parameters for the contact period, prior to beginning Level 0R processing of the contact period. LDTS must provide the following information.

Obtain (Extract the data transfer parameters):

- Valid_LDT_Parms(Num_Auth_Request, Timeout_Auth_Request, Num_Send_DAN_Attempt, Timeout_Send_DAN_Attempt, Timeout_Receive_DAA, Timeout_Receive_DDN, Num_Send_DDA_Attempt, Timeout_Send_DDA, Read_Sleep_Second)

2.2.1.6.4 Frequency

The MACS is expected to make available to the LDTS the contact period processing parameters prior to initiating Level 0R processing for the contact period. Modifications to the processing parameters cannot occur during Level 0R processing of any contact period on the associated LPS string.

2.2.1.6.5 Sizing

Table 2-17 specifies the sizing for the LDTS Processing Parameters interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing

Table 2-17. LDTS Processing Parameters Sizing

Interface Element	Number	Size (Bytes)
Valid_LDT_Parms Record	1	**30**

2.2.2 Error Reporting Thresholds

2.2.2.1 Description

The Error Reporting Thresholds interface contains all error reporting thresholds transferred from the MACS to the RDPS, MFPS, and PCDS subsystems.

2.2.2.2 RDPS

2.2.2.2.1 Format

The RDPS error reporting thresholds are detailed in table 2-18.

Table 2-18. RDPS Error Reporting Thresholds

Table	Attribute	Type	Range	Description
Valid_RDP_Thres	Sync_Thres	Number(8)	0-99999999 (Not NULL)	Modulo number of CADUs with synchronization errors allowed before operator notification
Valid_RDP_Thres	CRC_Thres	Number(8)	0-99999999 (Not NULL)	Modulo number of CADUs with CRC errors allowed before operator notification
Valid_RDP_Thres	RS_Thres	Number(8)	0-99999999 (Not NULL)	Modulo number of CADUs with Reed-Solomon errors allowed before operator notification
Valid_RDP_Thres	BCH_Thres	Number(8)	0-99999999 (Not NULL)	Modulo number of CADUs with BCH errors allowed before operator notification

2.2.2.2.2 IPC Mechanism

The IPC mechanism for the RDPS Error Reporting Thresholds interface is the LPS database. The database table name consisting of the error reporting thresholds is called "Valid_RDP_Thres". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

2.2.2.2.3 Data Transfer

The MACS and RDPS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Before the MACS initiates Level 0R processing, it may modify, on operator command, any or all of the RDPS error reporting thresholds. Within the record, the following may be modified.

Update (Modify the synchronization error threshold):

- Valid_RDP_Thres(Sync_Thres)

Update (Modify the CRC error threshold):

- Valid_RDP_Thres(CRC_Thres)

Update (Modify the Reed-Solomon EDAC threshold):

- Valid_RDP_Thres(RS_Thres)

Update (Modify the BCH EDAC threshold):

- Valid_RDP_Thres(BCH_Thres)

The MACS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database, and prior to initiating Level 0R processing.

The RDPS will extract the error reporting thresholds for the contact period, prior to beginning Level 0R processing of the contact period. RDPS must provide the following information.

Obtain (Extract the RDP thresholds):

- Valid_RDP_Thres(Sync_Thres, CRC_Thres, RS_Thres, BCH_Thres)

2.2.2.2.4 Frequency

The MACS is expected to make available to the RDPS the contact period error reporting thresholds prior to initiating Level 0R processing for the contact period. Modifications to the error reporting thresholds cannot occur during Level 0R processing of any contact period on the associated LPS string.

2.2.2.2.5 Sizing

Table 2-19 specifies the sizing for the RDPS Error Reporting Thresholds interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing

Table 2-19. RDPS Error Reporting Thresholds Sizing

Interface Element	Number	Size (Bytes)
Valid_RDP_Thres Record	1	**20**

2.2.2.3 MFPS

2.2.2.3.1 Format

The MFPS error reporting thresholds are detailed in table 2-20.

Table 2-20. MFPS Error Reporting Thresholds

Table	Attribute	Type	Range	Description
Valid_MFP_Thres	Mjf_CADU_Seq_Err_ Thres	Number(8)	0-99999999 (Not NULL)	Modulo number of ETM+ sequence counter errors allowed before operator notification
Valid_MFP_Thres	Scan_Dir_Thres	Number(8)	0-99999999 (Not NULL)	Modulo number of **TBD** errors allowed before operator notification
Valid_MFP_Thres	Mjf_Sync_Thres	Number(8)	0-99999999 (Not NULL)	Modulo number of ETM+ major frame synchronization errors allowed before operator notification
Valid_MFP_Thres	Mnf_Ctr_Thres	Number(8)	0-99999999 (Not NULL)	Modulo number of ETM+ minor frame errors allowed before operator notification
Valid_MFP_Thres	Eol_Thres	Number(8)	0-99999999 (Not NULL)	Modulo number of ETM+ end-of-line errors allowed before operator notification
Valid_MFP_Thres	Tc_Thres	Number(8)	0-99999999 (Not NULL)	Modulo number of ETM+ time code errors allowed before operator notification
Valid_MFP_Thres	Full_Mjf_Thres	Number(8)	0-99999999 (Not NULL)	Modulo number of entirely filled ETM+ major frames allowed before operator notification
Valid_MFP_Thres	Part_Mjf_Thres	Number(8)	0-99999999 (Not NULL)	Modulo number of partially filled ETM+ major frames allowed before operator notification

2.2.2.3.2 IPC Mechanism

The IPC mechanism for the MFPS Error Reporting Thresholds interface is the LPS database. The database table name consisting of the error reporting thresholds is called "Valid_MFP_Thres". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

2.2.2.3.3 Data Transfer

The MACS and MFPS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Before the MACS initiates Level 0R processing, it may modify, on operator command, any or all of the MFPS error reporting thresholds. Within the record, the following may be modified.

Update (Modify the ETM+ sequence counter error threshold):

- Valid_MFP_Thres(Mjf_CADU_Seq_Err_Thres)

Update (Modify the **TBD** error threshold):

- Valid_MFP_Thres(Scan_Dir_Thres)

Update (Modify the ETM+ major frame synchronization error threshold):

- Valid_MFP_Thres(Mjf_Sync_Thres)

Update (Modify the ETM+ minor frame error threshold):

- Valid_MFP_Thres(Mnf_Ctr_Thres)

Update (Modify the ETM+ end-of-line error threshold):

- Valid_MFP_Thres(Eol_Thres)

Update (Modify the ETM+ time code error threshold):

- Valid_MFP_Thres(Tc_Thres)

Update (Modify the entirely filled ETM+ major frame threshold):

- Valid_MFP_Thres(Full_Mjf_Thres)

Update (Modify the partially filled ETM+ major frames threshold):

- Valid_MFP_Thres(Part_Mjf_Thres)

The MACS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database, and prior to initiating Level 0R processing.

The MFPS will extract the error reporting thresholds for the contact period, prior to beginning Level 0R processing of the contact period. MFPS must provide the following information.

Obtain (Extract the MFP thresholds):

- Valid_MFP_Thres(Mjf_CADU_Seq_Err_Thres, Scan_Dir_Thres, Mjf_Sync_Thres, Mnf_Ctr_Thres, Eol_Thres, Tc_Thres, Full_Mjf_Thres, Part_Mjf_Thres)

2.2.2.3.4 Frequency

The MACS is expected to make available to the MFPS the contact period error reporting thresholds prior to initiating Level 0R processing for the contact period. Modifications to the error reporting thresholds cannot occur during Level 0R processing of any contact period on the associated LPS string.

2.2.2.3.5 Sizing

Table 2-21 specifies the sizing for the MFPS Error Reporting Thresholds interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing

Table 2-21. MFPS Error Reporting Thresholds Sizing

Interface Element	Number	Size (Bytes)
Valid_MFP_Thres Record	1	**40**

2.2.2.4 PCDS

2.2.2.4.1 Format

The PCDS error reporting thresholds are detailed in table 2-22.

Table 2-22. PCDS Error Reporting Thresholds

Table	Attribute	Type	Range	Description
Valid_PCD_Thres	Ephem_Position_Upper	Number(9,2)	-8388.60 < X < 8388.60 (Not NULL)	Largest valid ephemeris position data point (Km)
Valid_PCD_Thres	Ephem_Position_Lower	Number(9,2)	-8388.60 < X < 8388.60 (Not NULL)	Smallest valid ephemeris position data point (Km)
Valid_PCD_Thres	Ephem_Velocity_Upper	Number(3,2)	-8.00 < X < 8.00 (Not NULL)	Largest valid ephemeris velocity data point (Km/sec)
Valid_PCD_Thres	Ephem_Velocity_Lower	Number(3,2)	-8.00 < X < 8.00 (Not NULL)	Smallest valid ephemeris velocity data point (Km/sec)
Valid_PCD_Thres	Ephem_Crossproduct_ Max	Number(7,2)	53000.00 - 53200.00 (Not NULL) (**TBR**)	Largest valid cross product of ephemeris velocity and position

Table 2-22. PCDS Error Reporting Thresholds (Cont.)

Table	Attribute	Type	Range	Description
Valid_PCD_Thres	Ephem_Crossproduct_ Min	Number(7,2)	53000.00 - 53200.00 (Not NULL) (**TBR**)	Smallest valid cross product of ephemeris velocity and position
Valid_PCD_Thres	Attitude_Quaternion_Tol	Number(6,2)	0.00-9999.99 (Not NULL)	Modulo number of attitude quaternion errors allowed before operator notification
Valid_PCD_Thres	Num_Missing_Data_ Words	Number(8)	0-99999999 (Not NULL)	Modulo number of missing PCD words allowed before operator notification
Valid_PCD_Thres	Num_Failed_Votes	Number(8)	0-99999999 (Not NULL)	Modulo number of failed PCD majority votes allowed before operator notification

2.2.2.4.2 IPC Mechanism

The IPC mechanism for the PCDS Error Reporting Thresholds interface is the LPS database. The database table name consisting of the error reporting thresholds is called "Valid_PCD_Thres". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

2.2.2.4.3 Data Transfer

The MACS and PCDS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Before the MACS initiates Level 0R processing, it may modify, on operator command, any or all of the PCDS error reporting thresholds. Within the record, the following may be modified.

Update (Modify the largest valid ephemeris position data point (Km)):

- Valid_PCD_Thres(Ephem_Position_Upper)

Update (Modify the smallest valid ephemeris position data point (Km)):

- Valid_PCD_Thres(Ephem_Position_Lower)

Update (Modify the largest valid ephemeris velocity data point (Km/sec)):

- Valid_PCD_Thres(Ephem_Velocity_Upper)

Update (Modify the smallest valid ephemeris velocity data point (Km/sec)):

- Valid_PCD_Thres(Ephem_Velocity_Lower)

Update (Modify the largest valid cross product of ephemeris velocity and position):

- Valid_PCD_Thres(Ephem_Crossproduct_Max)

Update (Modify the smallest valid cross product of ephemeris velocity and position):

- Valid_PCD_Thres(Ephem_Crossproduct_Min)

Update (Modify the attitude quaternion error threshold):

- Valid_PCD_Thres(Attitude_Quaternion_Tol)

Update (Modify the missing PCD word threshold):

- Valid_PCD_Thres(Num_Missing_Data_Words)

Update (Modify the failed PCD majority vote threshold):

- Valid_PCD_Thres(Num_Failed_Votes)

The MACS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database, and prior to initiating Level 0R processing.

The PCDS will extract the error reporting thresholds for the contact period, prior to beginning Level 0R processing of the contact period. PCDS must provide the following information.

Obtain (Extract the PCD thresholds):

- Valid_PCD_Thres(Ephem_Position_Upper, Ephem_Position_Lower, Ephem_Velocity_Upper, Ephem_Velocity_Lower, Ephem_Crossproduct_Max, Ephem_Crossproduct_Min, Attitude_Quaternion_Tol, Num_Missing_Data_Words, Num_Failed_Votes)

2.2.2.4.4 Frequency

The MACS is expected to make available to the PCDS the contact period error reporting thresholds prior to initiating Level 0R processing for the contact period. Modifications to the error reporting thresholds cannot occur during Level 0R processing of any contact period on the associated LPS string.

2.2.2.4.5 Sizing

Table 2-23 specifies the sizing for the PCDS Error Reporting Thresholds interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing

Table 2-23. PCDS Error Reporting Thresholds Sizing

Interface Element	Number	Size (Bytes)
Valid_PCD_Thres Record	1	**42**

2.3 LPS Journal Messages

2.3.1 Description

The LPS journal messages consist of information necessary to monitor the LPS system. All LPS subsystems provide LPS messages to the MACS for logging to a journal file. The journal file provides a record of day-to-day operations of the LPS.

2.3.2 Format

The LPS Journal Messages are character strings providing information of varying degrees of severity. The message is accompanied by the unit name, the source line, and the severity or priority, number. The unit name is a character string, the source line is a numerical character string, and the priority number is a value between 0 and 7 as described in table 2-24.

Table 2-24. LPS Message Priorities

Number	Priority	Description
0	EMERG	EMERGENCY – the system is unusable
1	ALERT	ALERT – immediate action must be taken
2	CRIT	CRITICAL - critical condition
3	ERROR	ERROR - error condition
4	WARN	WARNING - warning condition
5	NOTICE	NOTICE - normal but significant
6	INFO	INFO - informational message
7	DEBUG	DEBUG - debug level messages intended for software troubleshooting, not of operational interest

2.3.3 IPC Mechanism

The IPC mechanism for the LPS Journal Messages interface will be via a global library macro called LPS_LOGMESSAGE. The macro is located in the global header file called "lps_prototype.h". Compile-time constants corresponding to the priority numbers are located in the global header file called "lps_constants.h".

2.3.4 Data Transfer

When an LPS subsystem generates an LPS journal message, it must use the LPS global library macro LPS_LOGMESSAGE to provide the message to the MACS. The usage is as follows.

```
LPS_LOGMESSAGE(Priority, Message)
```

Where *Priority* is the integer priority number as described in table 2-24, and *Message* is the actual character string message. The unit name and the source line, mentioned in section 2.3.2, are provided automatically by the LPS_LOGMESSAGE macro.

2.3.5 Frequency

The LPS Journal Messages will be provided to the MACS as the situation causing the message to be generated occurs.

2.3.6 Sizing

The LPS Journal Messages interface is limited to a text message size of 132 Bytes. This value is defined by the compile-time constant LPS_MSG_MAXLEN provided in the "lps_constants.h" header file.

2.4 Graphical User Interface

2.4.1 Description

The Graphical User Interface (GUI) interface describes all data provided by the LPS subsystems to the MACS necessary to support the GUI. This section is **TBD**.

Section 3. Metadata and Quality and Accounting

This section describes all information transferred between the RDPS and the MFPS.

3.1 Metadata Information

This subsection describes all metadata information transferred between the LPS subsystems and the MACS.

3.1.1 Description

The Metadata Information interface contains all key / search information and metadata information provided by the RDCS, MFPS, PCDS, and IDPS to the MACS. The subsystems are responsible for providing the MACS with metadata information at the contact level, the subinterval level, and at the scene level. The key / search information are not metadata information, but information required to identify the metadata information. The key / search information are listed in separate tables.

For the "Type" fields listed within the tables of the following subsections, an additional type has been added:

- SCTS - Spacecraft time string (Varchar2(30)) with 1/16th of a millisecond resolution, format "YYYY:DDD:HH:MM:SS.TTTTTTT" where
  - YYYY- Four digit year
  - DDD- Day of year from "001" to "366"
  - HH - Hour of day from "00" to "23"
  - MM - Minute of hour from "00" to "59"
  - SS - Second of minute from "00" to "59"
  - TTTTTTT - 1/16th of a millisecond represented in 7 fractional digits. Spacecraft clock increments are in 0.0000625 units.

3.1.2 RDCS

3.1.2.1 Format

The RDCS metadata key / search information are detailed in table 3-1.

Table 3-1. RDCS Metadata Key / Search Information

Table	Attribute	Type	Range	Description
RDC_Acct	Contact_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Contact sequence ID

The RDCS metadata information is detailed in table 3-2.

Table 3-2. RDCS Metadata Information

Table	Attribute	Type	Range	Description
RDC_Acct	Actual_Start_Time	Date	(Not NULL)	Contact period start time
RDC_Acct	Actual_Stop_Time	Date	(Not NULL)	Contact period stop time

3.1.2.2 IPC Mechanism

The Inter-Process Communication (IPC) mechanism for the RDCS Metadata Information interface is the LPS database. The database table name consisting of the metadata information is called "RDC_Acct". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

3.1.2.3 Data Transfer

The RDCS and MACS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

As the RDCS identifies a contact period, it creates a new database table record in RDC_Acct. Within the new record, the following must be initialized.

Set:

- RDC_Acct(Contact_Sequence_Id, Actual_Start_Time, Actual_Stop_Time)

Where:

- RDC_Acct(Contact_Sequence_Id) = A unique contact sequence identifier, and
- RDC_Acct(Actual_Start_Time) = actual contact period start time, and
- RDC_Acct(Actual_Stop_Time) = actual contact period stop time.

The RDCS will initialize a new table record for each contact period. The RDCS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database.

The MACS will extract the metadata information for the contact period after successful Level 0R processing of the contact period is complete. MACS must provide the following information.

Obtain (Extract the contact period information):

- RDC_Acct(Actual_Start_Time, Actual_Stop_Time)

Where:

- RDC_Acct(Contact_Sequence_Id) = Desired contact sequence ID.

3.1.2.4 Frequency

The RDCS is expected to make available to the MACS the contact period metadata information upon completion of data acquisition for the contact period.

3.1.2.5 Sizing

Table 3-3 specifies the sizing for the RDCS metadata interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing
- A quality and accounting retention time of 30 days (NOTE: Only 3-6 contacts will be on-line)
- A maximum of 6 contact periods per day

Table 3-3. RDCS Metadata Information Sizing

Interface Element	Number	Size (Bytes)
RDC_Acct Record (1 contact)	1	20
RDC_Acct Record (30 days)	180	**3,600**

3.1.3 MFPS

3.1.3.1 Format

The MFPS metadata key / search information are detailed in table 3-4.

Table 3-4. MFPS Metadata Key / Search Information

Table	Attribute	Type	Range	Description
LPS_File_Info	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
Sub_Intv	Contact_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Contact sequence ID
Sub_Intv	File_Version_Number	Number(10)	0-9 (Not NULL)	File version number
Sub_Intv	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
MFP_Acct	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
MFP_MJF_Acct	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
MFP_MJF_Acct	MFP_Mjf_Time	SCTS	(Not NULL)	Major frame time

The MFPS metadata information is detailed in table 3-5.

Table 3-5. MFPS Metadata Information

Table	Attribute	Type	Range	Description
LPS_File_Info	File_Name	Varchar2(256)	(Not NULL)	The output file name
LPS_File_Info	File_Type	Varchar2(3)	(Not NULL)	The L0R file type identifier
Sub_Intv	MF_Start_Time	SCTS	(Not NULL)	Subinterval start time
Sub_Intv	MF_Stop_Time	SCTS	(Not NULL)	Subinterval stop time

Table 3-5. MFPS Metadata Information (Cont.)

Table	Attribute	Type	Range	Description
Sub_Intv	VCID	Number(1)	1,2 (Not NULL)	Virtual channel ID
MFP_Acct	Mjf_Cnt	Number(8)	0-11725 (Not NULL)	Total number of ETM+ scans
MFP_MJF_Acct	CADUS_Received	Number(6)	0-999999 (Not NULL)	Total number of non-fill CADUS received for this major frame
MFP_MJF_Acct	Fly_Wheel_CADUS	Number(6)	0-999999 (Not NULL)	Total number of flywheel CADUS for this major frame
MFP_MJF_Acct	RS_Error_VCDUS	Number(6)	0-999999 (Not NULL)	Total number of Reed-Solomon errors for this major frame
MFP_MJF_Acct	BCH_Corr_VCDUS	Number(6)	0-999999 (Not NULL)	Total number of BCH corrected CADUS for this major frame
MFP_MJF_Acct	BCH_Uncorr_VCDUS	Number(6)	0-999999 (Not NULL)	Total number of uncorrectable CADUS for this major frame
MFP_MJF_Acct	Num_Bit_Errors (**TBR**)	Number(8)	0-99999999 (Not NULL)	Bit error count associated with this major frame. Used for Bit Error Rate (BER) calculation (**TBR**)
MFP_MJF_Acct	ETM_Timecode_Error_ Flag	Number(1)	LPS_TRUE, LPS_FALSE (Not NULL)	Indicates a time code error in this major frame
MFP_MJF_Acct	MJF_Filled_Flag	Char(1)	'E','P','N' (Not NULL)	Indicates the major frame fill: 'E' - Entirely filled, 'P' - Partially filled, 'N' - Not filled

3.1.3.2 IPC Mechanism

The IPC mechanism for the MFPS Metadata Information interface is the LPS database. The database table names consisting of the metadata information are called "LPS_File_Info", "MFP_Acct", "MFP_MJF_Acct", and "Sub_Intv". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

3.1.3.3 Data Transfer

The MFPS and MACS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Prior to completion of Level 0R processing, the MFPS must create a new database table record in MFP_MJF_Acct for each identified major frame, and a new database table record in MFP_Acct for each identified subinterval. The LPS_File_Info database table is detailed in section 9.2, and the Sub_Intv database table is detailed in section 6.2.4. Within the new records, the following must be initialized.

Set:

- LPS_File_Info(Sub_Intv_Sequence_Id, File_Name, File_Type)
- Sub_Intv(Contact_Sequence_Id, File_Version_Number, Sub_Intv_Sequence_Id, MF_Start_Time, MF_Stop_Time, VCID)
- MFP_Acct(Sub_Intv_Sequence_Id, Mjf_Cnt)
- MFP_MJF_Acct(Sub_Intv_Sequence_Id, MFP_Mjf_Time, CADUS_Received, Fly_Wheel_CADUS, RS_Error_VCDUS, BCH_Corr_VCDUS, BCH_Uncorr_VCDUS, Num_Bit_Errors (**TBR**), ETM_Timecode_Error_Flag, MJF_Filled_Flag)

Where:

- LPS_File_Info initialized as described in section 9.2, and
- Sub_Intv initialized as described in section 6.2.4, and
- MFP_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- MFP_Acct(Mjf_Cnt) = total number of ETM+ scans for the associated subinterval, and
- MFP_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- MFP_MJF_Acct(MFP_Mjf_Time) = associated major frame time, and
- MFP_MJF_Acct(CADUS_Received) = total number of non-fill CADUS received for this major frame, and
- MFP_MJF_Acct(Fly_Wheel_CADUS) = total number of flywheel CADUS for this major frame, and
- MFP_MJF_Acct(RS_Error_VCDUS) = total number of Reed-Solomon errors for this major frame, and
- MFP_MJF_Acct(BCH_Corr_VCDUS) = total number of BCH corrected CADUS for this major frame, and
- MFP_MJF_Acct(BCH_Uncorr_VCDUS) = total number of uncorrectable CADUS for this major frame, and
- MFP_MJF_Acct(Num_Bit_Errors) = bit error count associated with this major frame (**TBR**), and
- MFP_MJF_Acct(ETM_Timecode_Error_Flag) = set to LPS_TRUE to indicate a time code error in this major frame or LPS_FALSE to indicate no time code error in the major frame, and
- MFP_MJF_Acct(MJF_Filled_Flag) = set to indicate the major frame fill; values as described in table 3-5.

The MFPS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database.

The MACS will extract the metadata information for the contact period after successful Level 0R processing of the contact period is complete. MACS must provide the following information.

```
Obtain (Extract information for each subinterval in the contact period):
    •   Sub_Intv(Sub_Intv_Sequence_Id, MF_Start_Time, MF_Stop_Time, VCID)
Where:
    •   Sub_Intv(Contact_Sequence_Id) = Desired contact sequence ID, and
    •   Sub_Intv(File_Version_Number) = Desired file version number.
For each Sub_Intv_Sequence_Id returned from Sub_Intv:
    Obtain (Extract the subinterval MSCD and CAL file, and the ETM+ major frame subinterval accounting):
        •   LPS_File_Info(File_Name)
        •   MFP_Acct(Mjf_Cnt)
    Where:
        •   LPS_File_Info(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
        •   LPS_File_Info(File_Type) = "MSD" for the Mirror Scan Correction (MSCD) file, or
            "CAL" for the Calibration file, and
        •   MFP_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id.
    Obtain (Extract ETM+ scene accounting information for the subinterval):
        •   IDP_Acct(Scene_Center_Time, Scene_Start_Time, Scene_Stop_Time)
            See section 3.1.5 for details.
            NOTE: PCD_Scene_Acct(Scene_Center_Time) in conjunction with
            IDP_Acct(Scene_Start_Time, Scene_Stop_Time) will produce identical results.
    Where:
        •   IDP_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id
    For each Scene_Center_Time returned from IDP_Acct:
        Obtain (Extract the ETM+ major frame accounting for the scene):
            •   SUM(MFP_MJF_Acct.CADUS_Received), and
            •   SUM(MFP_MJF_Acct.Fly_Wheel_CADUS), and
            •   SUM(MFP_MJF_Acct.RS_Error_VCDUS), and
            •   SUM(MFP_MJF_Acct.BCH_Corr_VCDUS), and
            •   SUM(MFP_MJF_Acct.BCH_Uncorr_VCDUS), and
            •   SUM(MFP_MJF_Acct.Num_Bit_Errors) (TBR).
        Where:
            •   MFP_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
            •   MFP_MJF_Acct(MFP_Mjf_Time) >= IDP_Acct.Scene_Start_Time, and
            •   MFP_MJF_Acct(MFP_Mjf_Time) <= IDP_Acct.Scene_Stop_Time.
```

Obtain (Extract the number of ETM+ major frame timecode errors in the scene):

- COUNT(*) from MFP_MJF_Acct

Where:

- MFP_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- MFP_MJF_Acct(MFP_Mjf_Time) >= IDP_Acct.Scene_Start_Time, and
- MFP_MJF_Acct(MFP_Mjf_Time) <= IDP_Acct.Scene_Stop_Time, and
- MFP_MJF_Acct(ETM_Timecode_Error_Flag) = LPS_TRUE.

Obtain (Extract the number of entirely filled ETM+ major frames in the scene):

- COUNT(*) from MFP_MJF_Acct

Where:

- MFP_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- MFP_MJF_Acct(MFP_Mjf_Time) >= IDP_Acct.Scene_Start_Time, and
- MFP_MJF_Acct(MFP_Mjf_Time) <= IDP_Acct.Scene_Stop_Time, and
- MFP_MJF_Acct(MJF_Filled_Flag) = 'E'.

Obtain (Extract the number of partially filled ETM+ major frames in the scene):

- COUNT(*) from MFP_MJF_Acct

Where:

- MFP_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- MFP_MJF_Acct(MFP_Mjf_Time) >= IDP_Acct.Scene_Start_Time, and
- MFP_MJF_Acct(MFP_Mjf_Time) <= IDP_Acct.Scene_Stop_Time, and
- MFP_MJF_Acct(MJF_Filled_Flag) = 'P'.

3.1.3.4 Frequency

The MFPS is expected to make available to the MACS the subinterval and scene metadata information after MFPS completes, but before MACS completes Level 0R processing of the contact period.

3.1.3.5 Sizing

Table 3-6 specifies the sizing for the MFPS metadata interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing
- A metadata retention time of 1 contact period
- A maximum contact period of 14 minutes
- A maximum of 35 subintervals per contact (1 scene = 1 subinterval)
- A major frame time period of 71.343 msec (time between major frames)

Table 3-6. MFPS Metadata Information Sizing

Interface Element	Number	Size (Bytes)
LPS_File_Info Record	1	265
LPS_File_Info Record (1 contact)	35	**9,275**
Sub_Intv Record	1	80
Sub_Intv Record (1 contact)	35	**2,800**
MFP_Acct Record	1	11
MFP_Acct Record (1 contact)	35	**385**
MFP_MJF_Acct Record	1	64
MFP_MJF_Acct Record (1 contact)	11,775	**753,600**

3.1.4 PCDS

3.1.4.1 Format

The PCDS metadata key / search information are detailed in table 3-7.

Table 3-7. PCDS Metadata Key / Search Information

Table	Attribute	Type	Range	Description
LPS_File_Info	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
Bands_Present	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
Bands_Present	PCD_Cycle_Time	SCTS	(Not NULL)	Spacecraft time of the associated PCD cycle
PCD_Acct	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
PCD_MJF_Acct	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
PCD_Scene_Acct	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
PCD_Scene_Acct	Scene_Center_Time	SCTS	(Not NULL)	Actual WRS scene center time

The PCDS metadata information is detailed in table 3-8.

Table 3-8. PCDS Metadata Information

Table	Attribute	Type	Range	Description
LPS_File_Info	File_Name	Varchar2(256)	(Not NULL)	The output file name
LPS_File_Info	File_Type	Varchar2(3)	(Not NULL)	The L0R file type identifier
Bands_Present	Band_Present	Char(1)	(Not NULL)	Band states. Value stored as an 8 bit value, each bit representing the state of a band: 0 - Off, 1 - On Band bit positions per format: 1: 1 2 3 4 5 6 - - 2: - - - - - 6 7 P
PCD_Acct	First_PCD_MJF_Time	SCTS	(Not NULL)	Spacecraft time of the first PCD major frame in the associated subinterval
PCD_Acct	PCD_Stop_Time	SCTS	(Not NULL)	Spacecraft time of the last PCD major frame in the associated subinterval
PCD_Acct	Num_PCD_MJF	Number(3)	0-255 (Not NULL)	Total number of PCD major frames received in the associated subinterval
PCD_Acct	ETM_Last_On_Time	Number(16,7)	0.0-31622400.0 (Not NULL)	Last instrument on time
PCD_Acct	ETM_Last_Off_Time	Number(16,7)	0.0-31622400.0 (Not NULL)	Last instrument off time
PCD_Acct	UT1_Corrections	Number(6,5)	-0.90000 - 0.900000 (Not NULL)	UTC - UT1 in seconds from IAS Parameter file
PCD_Scene_Acct	WRS_Path_Nominal	Number(3)	1-233 (Not NULL)	WRS path number associated with the scene (The WRS path is the east to west index of the WRS table)
PCD_Scene_Acct	WRS_Row_Nominal	Number(3)	1-248 (Not NULL)	WRS row number associated with the scene (The WRS row is the north to south index of the WRS table)
PCD_Scene_Acct	Scene_Center_Lat	Number(7,4)	-90.0000 - +90.0000 (Not NULL)	Latitude of scene center
PCD_Scene_Acct	Scene_Center_Lon	Number(7,4)	-180.0000 - +180.0000 (Not NULL)	Longitude of scene center
PCD_Scene_Acct	Horizontal_Display_Shift	Number(8,4)	-9999 - 9999 (Not NULL)	Horizontal display shift for the associated scene
PCD_Scene_Acct	Scene_Upper_Left_Lat	Number(7,4)	-90.0000 - +90.0000 (Not NULL)	Latitude of upper left scene corner
PCD_Scene_Acct	Scene_Upper_Left_Lon	Number(7,4)	-180.0000 - +180.0000 (Not NULL)	Longitude of upper left scene corner
PCD_Scene_Acct	Scene_Upper_Right_Lat	Number(7,4)	-90.0000 - +90.0000 (Not NULL)	Latitude of upper right scene corner

Table 3-8. PCDS Metadata Information (Cont.)

Table	Attribute	Type	Range	Description
PCD_Scene_Acct	Scene_Upper_Right_Lon	Number(7,4)	-180.0000 - +180.0000 (Not NULL)	Longitude of upper right scene corner
PCD_Scene_Acct	Scene_Lower_Left_Lat	Number(7,4)	-90.0000 - +90.0000 (Not NULL)	Latitude of lower left scene corner
PCD_Scene_Acct	Scene_Lower_Left_Lon	Number(7,4)	-180.0000 - +180.0000 (Not NULL)	Longitude of lower left scene corner
PCD_Scene_Acct	Scene_Lower_Right_Lat	Number(7,4)	-90.0000 - +90.0000 (Not NULL)	Latitude of lower right scene corner
PCD_Scene_Acct	Scene_Lower_Right_Lon	Number(7,4)	-180.0000 - +180.0000 (Not NULL)	Longitude of lower right scene corner
PCD_Scene_Acct	Sun_Azimuth	Number(10,7)	-180.0000000 - 180.0000000 (Not NULL)	Solar angle associated with the scene
PCD_Scene_Acct	Sun_Elevation	Number(10,7)	-90.0000000 - 90.0000000 (Not NULL)	Solar elevation associated with the scene
PCD_Scene_Acct	Cal_Door_Activity_ Status	Number(1)	0,1 (Not NULL)	Indicates the ETM+ full aperture calibration activity during this scene
PCD_MJF_Acct	PCD_MJF_Time	SCTS	(Not NULL)	PCD major frame time for the associated PCD major frame
PCD_MJF_Acct	PCD_Words_Received	Number(6)	0-999999 (Not NULL)	Total number of PCD words, extracted from the unpacked PCD words, for the associated PCD major frame
PCD_MJF_Acct	Failed_PCD_Votes	Number(6)	0-999999 (Not NULL)	Total number of PCD words that encountered byte-voting errors during packing for the associated PCD major frame
PCD_MJF_Acct	Total_PCD_MNF	Number(3)	0-999 (Not NULL)	Total number of PCD minor frames constructed for the associated PCD major frame
PCD_MJF_Acct	Num_PCD_MNF_Sync_ Errors	Number(3)	0-999 (Not NULL)	Total number of PCD minor frames which encountered sync errors during their construction for the associated PCD major frame
PCD_MJF_Acct	Num_PCD_Filled_MNF	Number(3)	0-999 (Not NULL)	Total number of PCD minor frames which required data fill during their construction for the associated PCD major frame
PCD_MJF_Acct	PCD_Filled_MJF_Flag	Char(1)	'E','P','N' (Not NULL)	Indicates the PCD major frame fill: 'E' - Entirely filled, 'P' - Partially filled, 'N' - Not filled

Table 3-8. PCDS Metadata Information (Cont.)

Table	Attribute	Type	Range	Description
PCD_MJF_Acct	PCD_ADP_MJF_Flag	Char(1)	'G','R','M' (Not NULL)	Availability of spacecraft attitude data points (quaternation) received and processed for the associated PCD major frame: 'G' - Good, 'R' - Rejected, 'M' - Missing
PCD_MJF_Acct	PCD_EDP_MJF_Flag	Char(1)	'G','R','M' (Not NULL)	Availability of spacecraft ephemeris data point received and processed for the associated PCD major frame: 'G' - Good, 'R' - Rejected, 'M' - Missing

3.1.4.2 IPC Mechanism

The IPC mechanism for the PCDS Metadata Information interface is the LPS database. The database table names consisting of the metadata information are called "LPS_File_Info", "Bands_Present", "PCD_Acct", "PCD_Scene_Acct", and "PCD_MJF_Acct". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

3.1.4.3 Data Transfer

The PCDS and MACS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Prior to completion of Level 0R processing, the PCDS must create a new database table record in PCD_MJF_Acct for each identified PCD major frame, a new database table record in Bands_Present for each identified PCD cycle, a new database table record in PCD_Scene_Acct for each identified scene, and a new database table record in PCD_Acct for each identified subinterval. The LPS_File_Info database table is detailed in section 9.2. Within the new records, the following must be initialized.

Set:

- LPS_File_Info(Sub_Intv_Sequence_Id, File_Name, File_Type)
- Bands_Present(Sub_Intv_Sequence_Id, PCD_Cycle_Time, Band_Present)
- PCD_Acct(Sub_Intv_Sequence_Id, First_PCD_MJF_Time, PCD_Stop_Time, Num_PCD_MJF, ETM_Last_On_Time, ETM_Last_Off_Time, UT1_Corrections)

- PCD_MJF_Acct(Sub_Intv_Sequence_Id, PCD_MJF_Time, PCD_Words_Received, Failed_PCD_Votes, Total_PCD_MNF, Num_PCD_MNF_Sync_Errors, Num_PCD_Filled_MNF, PCD_Filled_MJF_Flag, PCD_ADP_MJF_Flag, PCD_EDP_MJF_Flag)
- PCD_Scene_Acct(Sub_Intv_Sequence_Id, Scene_Center_Time, WRS_Path_Nominal, WRS_Row_Nominal, Scene_Center_Lat, Scene_Center_Lon, Horizontal_Display_Shift, Scene_Upper_Left_Lat, Scene_Upper_Left_Lon, Scene_Upper_Right_Lat, Scene_Upper_Right_Lon, Scene_Lower_Left_Lat, Scene_Lower_Left_Lon, Scene_Lower_Right_Lat, Scene_Lower_Right_Lon, Sun_Azimuth, Sun_Elevation, Cal_Door_Activity_Status)

Where:

- LPS_File_Info initialized as described in section 9.2, and
- Bands_Present(Sub_Intv_Sequence_Id) = associated subinterval sequence ID, and
- Bands_Present(PCD_Cycle_Time) = associated PCD cycle time, and
- Bands_Present(Band_Present) = band states as described in table 3-8, and
- PCD_Acct(Sub_Intv_Sequence_Id) = associated subinterval sequence ID, and
- PCD_Acct(First_PCD_MJF_Time) = spacecraft time of the first PCD major frame in the associated subinterval, and
- PCD_Acct(PCD_Stop_Time) = spacecraft time of the last PCD major frame in the associated subinterval, and
- PCD_Acct(Num_PCD_MJF) = total number of PCD major frames received in the associated subinterval, and
- PCD_Acct(ETM_Last_On_Time) = last instrument on time as described in Specification Document 7, and
- PCD_Acct(ETM_Last_Off_Time) = last instrument off time as described in Specification Document 7, and
- PCD_Acct(UT1_Corrections) = UT1 from the IAS Parameter file in seconds, and
- PCD_MJF_Acct(Sub_Intv_Sequence_Id) = associated subinterval sequence ID, and
- PCD_MJF_Acct(PCD_MJF_Time) = associated PCD major frame time, and
- PCD_MJF_Acct(PCD_Words_Received) = total number of PCD words, extracted from the unpacked PCD words, for the associated PCD major frame, and
- PCD_MJF_Acct(Failed_PCD_Votes) = total number of PCD words that encountered byte-voting errors during packing for the associated PCD major frame, and
- PCD_MJF_Acct(Total_PCD_MNF) = total number of PCD minor frames constructed for the associated PCD major frame, and
- PCD_MJF_Acct(Num_PCD_MNF_Sync_Errors) = total number of PCD minor frames which encountered synchronization errors during their construction for the associated PCD major frame, and
- PCD_MJF_Acct(Num_PCD_Filled_MNF) = total number of PCD minor frames which required data fill during their construction for the associated PCD major frame, and

- PCD_MJF_Acct(PCD_Filled_MJF_Flag) = indicates the PCD major frame fill; values described in table 3-8, and
- PCD_MJF_Acct(PCD_ADP_MJF_Flag) = availability of spacecraft attitude data points (quaternation) received and processed for the associated PCD major frame; values described in table 3-8, and
- PCD_MJF_Acct(PCD_EDP_MJF_Flag) = availability of spacecraft ephemeris data point received and processed for the associated PCD major frame; values described in table 3-8, and
- PCD_Scene_Acct(Sub_Intv_Sequence_Id) = associated subinterval sequence ID, and
- PCD_Scene_Acct(Scene_Center_Time) = calculated scene center time for the associated scene, and
- PCD_Scene_Acct(WRS_Path_Nominal) = WRS path number associated with the scene, and
- PCD_Scene_Acct(WRS_Row_Nominal) = WRS row number associated with the scene (The WRS row is the north to south index of the WRS table), and
- PCD_Scene_Acct(Scene_Center_Lat) = latitude of scene center, and
- PCD_Scene_Acct(Scene_Center_Lon) = longitude of scene center, and
- PCD_Scene_Acct(Horizontal_Display_Shift) = horizontal display shift for the associated scene, and
- PCD_Scene_Acct(Scene_Upper_Left_Lat) = latitude of upper left scene corner, and
- PCD_Scene_Acct(Scene_Upper_Left_Lon) = longitude of upper left scene corner, and
- PCD_Scene_Acct(Scene_Upper_Right_Lat) = latitude of upper right scene corner, and
- PCD_Scene_Acct(Scene_Upper_Right_Lon) = longitude of upper right scene corner, and
- PCD_Scene_Acct(Scene_Lower_Left_Lat) = latitude of lower left scene corner, and
- PCD_Scene_Acct(Scene_Lower_Left_Lon) = longitude of lower left scene corner, and
- PCD_Scene_Acct(Scene_Lower_Right_Lat) = latitude of lower right scene corner, and
- PCD_Scene_Acct(Scene_Lower_Right_Lon) = longitude of lower right scene corner, and
- PCD_Scene_Acct(Sun_Azimuth) = solar angle associated with the scene, and
- PCD_Scene_Acct(Sun_Elevation) = solar elevation associated with the scene, and
- PCD_Scene_Acct(Cal_Door_Activity_Status) = indicates the ETM+ full aperture calibration activity during this scene.

The PCDS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database.

The MACS will extract the metadata information for the contact period after successful Level 0R processing of the contact period is complete. MACS must provide the following information for the contact period.

```
Obtain (Extract information for each subinterval in the contact period):
    •   Sub_Intv(Sub_Intv_Sequence_Id, MF_Start_Time, MF_Stop_Time)
Where:
    •   Sub_Intv(Contact_Sequence_Id) = Desired contact sequence ID, and
    •   Sub_Intv(File_Version_Number) = Desired file version number.
For each Sub_Intv_Sequence_Id returned from Sub_Intv:
    Obtain (Extract the subinterval PCD file, and the PCD subinterval accounting):
        •   LPS_File_Info(File_Name)
        •   PCD_Acct(First_PCD_MJF_Time, PCD_Stop_Time, Num_PCD_MJF, ETM_Last_On_Time,
            ETM_Last_Off_Time, UT1_Corrections)
    Where:
        •   LPS_File_Info(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
        •   LPS_File_Info(File_Type) = "PCD" for the Payload Correction Data (PCD) file, and
        •   PCD_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id
    Obtain (Extract the starting WRS row for the subinterval):
        •   PCD_Scene_Acct(WRS_Row_Nominal)
    Where:
        •   PCD_Scene_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
        •   Scene_Center_Time IN
        Obtain (Locate the earliest scene center time record):
            •   MIN(PCD_Scene_Acct.Scene_Center_Time)
        Where:
            •   PCD_Scene_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id.
    Obtain (Extract the ending WRS row and the WRS path for the subinterval):
        •   PCD_Scene_Acct(WRS_Row_Nominal, WRS_Path_Nominal)
    Where:
        •   PCD_Scene_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
        •   PCD_Scene_Acct.Scene_Center_Time IN
        Obtain (Locate the latest scene center time record):
            •   MAX(PCD_Scene_Acct.Scene_Center_Time)
        Where:
            •   PCD_Scene_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id.
```

Obtain (Extract the band states from the first PCD cycle in the subinterval):

- Bands_Present(Band_Present)

Where:

- Bands_Present(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- PCD_Cycle_Time IN

  Obtain (Locate the earliest PCD Cycle time record):

  - MIN(Bands_Present.PCD_Cycle_Time)

  Where:

  - Bands_Present(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
  - Bands_Present(PCD_Cycle_Time) >= Sub_Intv.MF_Start_Time, and
  - Bands_Present(PCD_Cycle_Time) <= Sub_Intv.MF_Stop_Time.

Obtain (Extract PCD scene accounting information for the subinterval):

- IDP_Acct(Scene_Center_Time, Scene_Start_Time Scene_Stop_Time)
  See section 3.1.5 for details.
  NOTE: PCD_Scene_Acct(Scene_Center_Time) in conjunction with
  IDP_Acct(Scene_Start_Time, Scene_Stop_Time) will produce identical results.

Where:

- IDP_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id

For each Scene_Center_Time returned from IDP_Acct:

  Obtain (Extract the PCD major frame accounting for the scene):

  - SUM(PCD_MJF_Acct.PCD_Words_Received), and
  - SUM(PCD_MJF_Acct.Failed_PCD_Votes), and
  - SUM(PCD_MJF_Acct.Total_PCD_MNF), and
  - SUM(PCD_MJF_Acct.Num_PCD_MNF_Sync_Errors), and
  - SUM(PCD_MJF_Acct.Num_PCD_Filled_MNF)

  Where:

  - PCD_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
  - PCD_MJF_Acct(PCD_MJF_Time) >= IDP_Acct.Scene_Start_Time, and
  - PCD_MJF_Acct(PCD_MJF_Time) <= IDP_Acct.Scene_Stop_Time.

Obtain (Extract the number of non-filled PCD major frames in the scene):

- COUNT(*) from PCD_MJF_Acct

Where:

- PCD_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- PCD_MJF_Acct(PCD_MJF_Time) >= IDP_Acct.Scene_Start_Time, and
- PCD_MJF_Acct(PCD_MJF_Time) <= IDP_Acct.Scene_Stop_Time, and
- PCD_MJF_Acct(PCD_Filled_MJF_Flag) = 'N'.

Obtain (Extract the number of good attitude data points in the scene):

- COUNT(*) from PCD_MJF_Acct

Where:

- PCD_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- PCD_MJF_Acct(PCD_MJF_Time) >= IDP_Acct.Scene_Start_Time, and
- PCD_MJF_Acct(PCD_MJF_Time) <= IDP_Acct.Scene_Stop_Time, and
- PCD_MJF_Acct(PCD_ADP_MJF_Flag) = 'G'.

Obtain (Extract the number of rejected attitude data points in the scene):

- COUNT(*) from PCD_MJF_Acct

Where:

- PCD_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- PCD_MJF_Acct(PCD_MJF_Time) >= IDP_Acct.Scene_Start_Time, and
- PCD_MJF_Acct(PCD_MJF_Time) <= IDP_Acct.Scene_Stop_Time, and
- PCD_MJF_Acct(PCD_ADP_MJF_Flag) = 'R'.

Obtain (Extract the number of missing attitude data points in the scene):

- COUNT(*) from PCD_MJF_Acct

Where:

- PCD_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- PCD_MJF_Acct(PCD_MJF_Time) >= IDP_Acct.Scene_Start_Time, and
- PCD_MJF_Acct(PCD_MJF_Time) <= IDP_Acct.Scene_Stop_Time, and
- PCD_MJF_Acct(PCD_ADP_MJF_Flag) = 'M'.

Obtain (Extract the number of good ephemeris data points in the scene):

- COUNT(*) from PCD_MJF_Acct

Where:

- PCD_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- PCD_MJF_Acct(PCD_MJF_Time) >= IDP_Acct.Scene_Start_Time, and
- PCD_MJF_Acct(PCD_MJF_Time) <= IDP_Acct.Scene_Stop_Time, and
- PCD_MJF_Acct(PCD_EDP_MJF_Flag) = 'G'.

Obtain (Extract the number of rejected ephemeris data points in the scene):

- COUNT(*) from PCD_MJF_Acct

Where:

- PCD_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- PCD_MJF_Acct(PCD_MJF_Time) >= IDP_Acct.Scene_Start_Time, and
- PCD_MJF_Acct(PCD_MJF_Time) <= IDP_Acct.Scene_Stop_Time, and
- PCD_MJF_Acct(PCD_EDP_MJF_Flag) = 'R'.

Obtain (Extract the number of missing ephemeris data points in the scene):

- COUNT(*) from PCD_MJF_Acct

Where:

- PCD_MJF_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- PCD_MJF_Acct(PCD_MJF_Time) >= IDP_Acct.Scene_Start_Time, and
- PCD_MJF_Acct(PCD_MJF_Time) <= IDP_Acct.Scene_Stop_Time, and
- PCD_MJF_Acct(PCD_EDP_MJF_Flag) = 'M'.

Obtain (Extract PCD scene accounting information for the subinterval):

- PCD_Scene_Acct(Scene_Center_Time)

Where:

- PCD_Scene_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id

For each Scene_Center_Time returned from PCD_Scene_Acct:

Obtain (Extract the PCD accounting for the scene):

- PCD_Scene_Acct(WRS_Path_Nominal, WRS_Row_Nominal, Scene_Center_Lat, Scene_Center_Lon, Horizontal_Display_Shift, Scene_Upper_Left_Lat, Scene_Upper_Left_Lon, Scene_Upper_Right_Lat, Scene_Upper_Right_Lon, Scene_Lower_Left_Lat, Scene_Lower_Left_Lon, Scene_Lower_Right_Lat, Scene_Lower_Right_Lon, Sun_Azimuth, Sun_Elevation, Cal_Door_Activity_Status)

Where:

- PCD_Scene_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- PCD_Scene_Acct(Scene_Center_Time) = PCD_Scene_Acct.Scene_Center_Time, and

3.1.4.4 Frequency

The PCDS is expected to make available to the MACS the subinterval and scene metadata information upon completion of Level 0R processing of the contact period by the PCDS.

3.1.4.5 Sizing

Table 3-9 specifies the sizing for the PCDS metadata interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing
- A metadata retention time of 1 contact period
- A maximum of 35 subintervals per contact
- A maximum of 35 scenes per contact
- A maximum of 52 PCD cycles per contact (2 partial scenes)
- A maximum of 207 PCD major frames per contact (2 partial scenes)

Table 3-9. PCDS Metadata Information Sizing

Interface Element	Number	Size (Bytes)
LPS_File_Info Record	1	265
LPS_File_Info Record (1 contact)	35	**9,275**
Bands_Present Record	1	37
Bands_Present Record (1 contact)	52	**1,924**
PCD_Acct Record	1	94
PCD_Acct Record (1 contact)	35	**3,290**
PCD_MJF_Acct Record	1	56
PCD_MJF_Acct Record (1 contact)	207	**11,592**
PCD_Scene_Acct Record	1	111
PCD_Scene_Acct Record (1 contact)	35	**3,885**

3.1.5 IDPS

3.1.5.1 Format

The IDPS metadata key / search information are detailed in table 3-10.

Table 3-10. IDPS Metadata Key / Search Information

Table	Attribute	Type	Range	Description
LPS_File_Info	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
Band_Gain_States	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
Band_Gain_States	Scene_Center_Time	SCTS	(Not NULL)	Actual WRS scene center time
IDP_Acct	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
IDP_Acct	Scene_Center_Time	SCTS	(Not NULL)	Actual WRS scene center time

The IDPS metadata information is detailed in table 3-11.

Table 3-11. IDPS Metadata Information

Table	Attribute	Type	Range	Description
LPS_File_Info	File_Name	Varchar2(256)	(Not NULL)	The output file name
LPS_File_Info	File_Type	Varchar2(3)	(Not NULL)	The L0R file type identifier
Band_Gain_States	Band	Number(1)	1-8 (Not NULL)	Band number

Band_Gain_States	Band_Gain	Char(1)	'L','H'
			(Not NULL)
				Band gain associated with the band (one record for each band):
				`'L' - Low gain,`
				`'H' - High gain`

Band_Gain_States	Band_Gain_Change	Char(1)	'0','+','-'
			(Not NULL)
				Indicates the band gain change within a scene (one record for each band):
				`'0' - No change,`
				`'+' - Low to high,`
				`'-' - High to low`

IDP_Acct	Scene_Start_Time	SCTS	(Not NULL)	Measured WRS scene start time
IDP_Acct	Scene_Stop_Time	SCTS	(Not NULL)	Measured WRS scene end time
IDP_Acct	Sub_Intv_Scene_Number	Number(2)	1-99 (Not NULL)	Scene number within the subinterval
IDP_Acct	Scene_Center_Scan_ Number	Number(5)	-175 - 11725 (Not NULL)	Scan number, or major frame number, whose center is closest to the center of a nominal WRS scene
IDP_Acct	CCA_Quadrant1_Score	Number(5,2)	0.00-100.00	1st quadrant cloud coverage (%)

Table 3-11. IDPS Metadata Information (Cont.)

Table	Attribute	Type	Range	Description
IDP_Acct	CCA_Quadrant2_Score	Number(5,2)	0.00-100.00	2nd quadrant cloud coverage (%)
IDP_Acct	CCA_Quadrant3_Score	Number(5,2)	0.00-100.00	3rd quadrant cloud coverage (%)
IDP_Acct	CCA_Quadrant4_Score	Number(5,2)	0.00-100.00	4th quadrant cloud coverage (%)
IDP_Acct	CCA_Aggregate_Score	Number(5,2)	0.00-100.00	Scene cloud coverage (%)
IDP_Acct	Full_or_Partial_Scene	Char(1)	'F','P' (Not NULL)	Flag identifying that the associated scene is full or partial: 'F' - Full scene, 'P' - Partial scene
Valid_Band_Parms	CCA_Method	Varchar2(256)	(Not NULL)	Indicates the cloud cover method used for cloud cover assessment

3.1.5.2 IPC Mechanism

The IPC mechanism for the IDPS Metadata Information interface is the LPS database. The database table names consisting of the metadata information are called "LPS_File_Info", "Band_Gain_States", "IDP_Acct", and "Valid_Band_Parms". Valid_Band_Parms is a single record database table. The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

3.1.5.3 Data Transfer

The IDPS and MACS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Prior to completion of Level 0R processing, the IDPS must create a new database table record in Band_Gain_States and IDP_Acct for each identified scene. The LPS_File_Info database table is detailed in section 9.2. Within the new records, the following must be initialized.

Set:

- LPS_File_Info(Sub_Intv_Sequence_Id, File_Name, File_Type)
- Band_Gain_States(Sub_Intv_Sequence_Id, Scene_Center_Time, Band, Band_Gain, Band_Gain_Change)
- IDP_Acct(Sub_Intv_Sequence_Id, Scene_Center_Time, Sub_Intv_Scene_Number, Scene_Center_Scan_Number, CCA_Quadrant1_Score, CCA_Quadrant2_Score, CCA_Quadrant3_Score, CCA_Quadrant4_Score, CCA_Aggregate_Score, Full_or_Partial_Scene)
- Valid_Band_Parms(CCA_Method)

Where:

- LPS_File_Info initialized as described in section 9.2, and
- Band_Gain_States(Sub_Intv_Sequence_Id) = associated subinterval sequence ID, and
- Band_Gain_States(Scene_Center_Time) = scene center time associated with the band, and
- Band_Gain_States(Band) = associated band number, and
- Band_Gain_States(Band_Gain) = gain of the associated band, and
- Band_Gain_States(Band_Gain_Change) = value indicating the band gain change within the associated scene (one record for each band); values described in table 3-11, and
- IDP_Acct(Sub_Intv_Sequence_Id) = associated subinterval sequence ID, and
- IDP_Acct(Scene_Center_Time) = associated scene center time, and
- IDP_Acct(Scene_Start_Time) = measured WRS scene start time, and
- IDP_Acct(Scene_Stop_Time) = measured WRS scene stop time, and
- IDP_Acct(Sub_Intv_Scene_Number) = unique scene number within the subinterval, and
- IDP_Acct(Scene_Center_Scan_Number) = scan number, or major frame number, whose center is closest to the center of a nominal WRS scene, and
- IDP_Acct(CCA_Quadrant1_Score) = percentage of 1st quadrant cloud coverage, and
- IDP_Acct(CCA_Quadrant2_Score) = percentage of 2nd quadrant cloud coverage, and
- IDP_Acct(CCA_Quadrant3_Score) = percentage of 3rd quadrant cloud coverage, and
- IDP_Acct(CCA_Quadrant4_Score) = percentage of 4th quadrant cloud coverage, and
- IDP_Acct(CCA_Aggregate_Score) = percentage of scene cloud coverage, and
- IDP_Acct(Full_or_Partial_Scene) = flag identifying that the associated scene is full or partial: values described in table 3-11, and

The IDPS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database.

The MACS will extract the metadata information for the contact period after successful Level 0R processing of the contact period is complete. MACS must provide the following information for the contact period.

Obtain (Extract information for each subinterval in the contact period):

- Sub_Intv(Sub_Intv_Sequence_Id, MF_Start_Time, MF_Stop_Time)

Where:

- Sub_Intv(Contact_Sequence_Id) = Desired contact sequence ID, and
- Sub_Intv(File_Version_Number) = Desired file version number.

```
For each Sub_Intv_Sequence_Id returned from Sub_Intv:
    For each band file (1-8):
        For each band file segment (1-4):
            Obtain (Extract the Band file information):
                •   LPS_File_Info(File_Name)
            Where:
                •   LPS_File_Info(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
                •   LPS_File_Info(File_Type) = "B" concatenated with the band number, concatenated
                    with the segment number to obtain the Band File.  See table 9-13 for more details
    Obtain (Extract the total number of scenes for the subinterval):
        •   COUNT(*) from IDP_Acct
    Where:
        •   IDP_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id
    Obtain (Extract the total number of partial scenes for the subinterval):
        •   COUNT(*) from IDP_Acct
    Where:
        •   IDP_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
        •   IDP_Acct.Full_or_Partial_Scene = 'P'.
    Obtain (Extract ETM+ scene accounting information for the subinterval):
        •   IDP_Acct(Scene_Center_Time)
    Where:
        •   IDP_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id
    For each Scene_Center_Time returned from IDP_Acct:
        Obtain (Extract the cloud cover method for the scene):
            •   Valid_Band_Parms(CCA_Method)
        Obtain (Extract the scene Browse file, and the scene accounting):
            •   LPS_File_Info(File_Name)
            •   IDP_Acct(Sub_Intv_Scene_Number, Scene_Center_Scan_Number,
                CCA_Quadrant1_Score, CCA_Quadrant2_Score, CCA_Quadrant3_Score,
                CCA_Quadrant4_Score, CCA_Aggregate_Score, Full_or_Partial_Scene)
        Where:
            • LPS_File_Info(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
            • LPS_File_Info(File_Type) = "R" concatenated with IDP_Acct.Sub_Intv_Scene_Number
                for the Browse file, and
            • IDP_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
            • IDP_Acct(Scene_Center_Time) = IDP_Acct.Scene_Center_Time.
    Obtain (Extract the ETM+ band gain state information for each scene):
        • Band_Gain_States(Scene_Center_Time)
    Where:
        • Band_Gain_States(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id
    For each Scene_Center_Time returned from Band_Gain_States:
        For each band (1-8):
            Obtain (Extract the band gain information for each band):
                • Band_Gain_States(Band_Gain, Band_Gain_Change)
            Where:
                • Band_Gain_States(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
                • Band_Gain_States(Scene_Center_Time) = Band_Gain_States.Scene_Center_Time,
                    and
                • Band_Gain_States(Band) = current band.
```

3.1.5.4 Frequency

The IDPS is expected to make available to the MACS the subinterval and scene metadata information upon completion of Level 0R processing of the contact period by the IDPS.

3.1.5.5 Sizing

Table 3-12 specifies the sizing for the IDPS metadata interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing
- A metadata retention time of 1 contact period
- A maximum of 35 scenes per contact

Table 3-12. IDPS Metadata Information Sizing

Interface Element	Number	Size (Bytes)
LPS_File_Info Record	1	265
LPS_File_Info Record (1 contact)	35	**9,275**
Band_Gain_States Record	1	40
Band_Gain_States Record (1 contact)	35	**1,400**
IDP_Acct Record	1	123
IDP_Acct Record (1 contact)	35	**4,305**
Valid_Band_Parms Record	1	**256**

3.2 Quality and Accounting Information

This subsection describes all quality and accounting information transferred between the LPS subsystems and the MACS.

3.2.1 Description

The Quality and Accounting Information interface contains all key / search information and quality and accounting information provided by the RDCS, RDPS, and MFPS to the MACS for the LPS quality and accounting report. The subsystems are responsible for providing the MACS with quality and accounting information at the contact level and subinterval level. The key / search information are not quality and accounting information, but information required to identify the quality and accounting information. The key / search information are listed in separate tables.

For the "Type" fields listed within the tables of the following subsections, an additional type has been added:

- SCTS - Spacecraft time string (Varchar2(30)) with 1/16th of a millisecond resolution, format "YYYY:DDD:HH:MM:SS.TTTTTTT" where
  - YYYY- Four digit year
  - DDD- Day of year from "001" to "366"
  - HH - Hour of day from "00" to "23"
  - MM - Minute of hour from "00" to "59"
  - SS - Second of minute from "00" to "59"
  - TTTTTTT - 1/16th of a millisecond represented in 7 fractional digits. Spacecraft clock increments are in 0.0000625 units.

3.2.2 RDCS

3.2.2.1 Format

The RDCS quality and accounting key / search information are detailed in table 3-13.

Table 3-13. RDCS Quality and Accounting Key / Search Information

Table	Attribute	Type	Range	Description
RDC_Acct	Contact_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Contact sequence ID

The RDCS quality and accounting information are detailed in table 3-14.

Table 3-14. RDCS Quality and Accounting Information

Table	Attribute	Type	Range	Description
RDC_Acct	Raw_Data_File_Name (**TBR**)	Varchar2(512)	(Not NULL)	Raw data filename including the path for the associated contact period
RDC_Acct	LPS_Hardware_String_Id	Varchar2(20)	(Not NULL)	LPS string ID that captured the contact
RDC_Acct	Received_Data_Vol	Number(7,2)	0.00-99999.99 (Not NULL)	Actual file size of the raw data capture file associated with the contact period (MBytes)
RDC_Acct	Actual_Start_Time	Date	(Not NULL)	Contact period start time
RDC_Acct	Actual_Stop_Time	Date	(Not NULL)	Contact period stop time

3.2.2.2 IPC Mechanism

The IPC mechanism for the RDCS Quality and Accounting Information interface is the LPS database. The database table name consisting of the quality and accounting information is called "RDC_Acct". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

3.2.2.3 Data Transfer

The RDCS and MACS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

As the RDCS identifies a contact period, it creates a new database table record in RDC_Acct. Within the new record, the following must be initialized.

Set:

- RDC_Acct(Contact_Sequence_Id, Raw_Data_File_Name (**TBR**), LPS_Hardware_String_Id, Received_Data_Vol, Actual_Start_Time, Actual_Stop_Time)

Where:

- RDC_Acct(Contact_Sequence_Id) = A unique contact sequence identifier, and
- RDC_Acct(Raw_Data_File_Name) = the raw data capture filename (**TBR**), and
- RDC_Acct(LPS_Hardware_String_Id) = the LPS hardware string ID that captured the raw data, e.g. lps001, and
- RDC_Acct(Received_Data_Vol) = the actual size of the raw data capture file in MBytes, and
- RDC_Acct(Actual_Start_Time) = the actual start time of the captured data, and
- RDC_Acct(Actual_Stop_Time) = the actual stop time of the captured data.

The RDCS will initialize a new table record for each contact period. The RDCS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database.

The MACS will extract the quality and accounting information for the contact period, upon operator command, after successful Level 0R processing of the contact period is complete. MACS must provide the following information.

Obtain (Extract the contact period Q/A information):

- RDC_Acct(Raw_Data_File_Name (**TBR**), LPS_Hardware_String_Id, Received_Data_Vol, Actual_Start_Time, Actual_Stop_Time)

Where:

- RDC_Acct(Contact_Sequence_Id) = Desired contact sequence ID.

3.2.2.4 Frequency

The RDCS is expected to make available to the MACS the contact period quality and accounting information before Level 0R processing of the contact period.

3.2.2.5 Sizing

Table 3-15 specifies the sizing for the RDCS Quality and Accounting interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing
- A quality and accounting retention time of 30 days
- A maximum of 6 contact periods per day

Table 3-15. RDCS Quality and Accounting Information Sizing

Interface Element	Number	Size (Bytes)
RDC_Acct Record	1	557
RDC_Acct Record (30 days)	180	**100,260**

3.2.3 RDPS

3.2.3.1 Format

The RDPS quality and accounting key / search information are detailed in table 3-16.

Table 3-16. RDPS Quality and Accounting Key / Search Information

Table	Attribute	Type	Range	Description
RDP_Acct	Contact_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Contact sequence ID
RDP_Acct	File_Version_Number	Number(10)	0-9 (Not NULL)	File version number
RDP_Acct	CCSDS_Parms_Id	Number(10)	0-ULONG_MAX (Not NULL)	CCSDS parameter identifier from the Valid_CCSDS_Parms
Valid_CCSDS_Parms	CCSDS_Parms_Id	Number(10)	1-ULONG_MAX (Not NULL)	CCSDS parameters sequence ID

The RDPS quality and accounting information are detailed in table 3-17.

Table 3-17. RDPS Quality and Accounting Information

Table	Attribute	Type	Range	Description
RDP_Acct	Inverted_CADU_Cnt	Number(8)	0-99999999 (Not NULL)	Count of CADUs received with inverted polarity
RDP_Acct	Polarity_Change_Cnt	Number(8)	0-99999999 (Not NULL)	Number of polarity changes in the CADUs
RDP_Acct	CADU_Bit_Slip_Cnt	Number(8)	0-99999999 (Not NULL)	Number of CADUs containing bit slips
RDP_Acct	CADU_Sync_Err_Cnt	Number(8)	0-99999999 (Not NULL)	Number of CADUs containing sync errors
RDP_Acct	CADU_Rcv_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of received CADUs
RDP_Acct	CADU_Flywheel_Cnt	Number(8)	0-99999999 (Not NULL)	Number of flywheel CADUs
RDP_Acct	Fill_CADU_Cnt (**TBR**)	Number(8)	0-99999999 (Not NULL)	Number of fill CADUs
RDP_Acct	VCDU_Header1_Corr_ Err_Cnt	Number(8)	0-99999999 (Not NULL)	Number of virtual channel 1 VCDUs with correctable VCDU headers
RDP_Acct	VCDU_Header2_Corr_ Err_Cnt	Number(8)	0-99999999 (Not NULL)	Number of virtual channel 2 VCDUs with correctable VCDU headers
RDP_Acct	VCDU_Header_Uncorr_ Err_Cnt	Number(8)	0-99999999 (Not NULL)	Number of uncorrectable VCDU headers
RDP_Acct	BCH_Data_Corrected_ Err_Cnt	Number(8)	0-99999999 (Not NULL)	Number of CADUs with corrected BCH errors in the mission data zone
RDP_Acct	BCH_Data_Uncorrected_ Err_Cnt	Number(8)	0-99999999 (Not NULL)	Number of CADUs with uncorrected BCH errors in the mission data zone
RDP_Acct	BCH_Data_Corrected_ Bits_Cnt (**TBR**)	Number(8)	0-99999999 (Not NULL)	Number of bits BCH corrected in the mission data zone
RDP_Acct	BCH_Ptr_Corrected_ Err_Cnt	Number(8)	0-99999999 (Not NULL)	Number of CADUs with corrected BCH errors in the data pointer area
RDP_Acct	BCH_Ptr_Uncorrected_ Err_Cnt	Number(8)	0-99999999 (Not NULL)	Number of CADUs with uncorrected BCH errors in the data pointer area
RDP_Acct	BCH_Ptr_Corrected_ Bits_Cnt (**TBR**)	Number(8)	0-99999999 (Not NULL)	Number of bits BCH corrected in the data pointer area
RDP_Acct	CADU_CRC_Err_Cnt	Number(8)	0-99999999 (Not NULL)	Number of CADUs with CRC errors
Valid_CCSDS_Parms	Insertion_Time	Date	(Not NULL)	Record insertion time
Valid_CCSDS_Parms	CADU_Search_Tol	Number(1)	1-3 (Not NULL)	Search tolerance parameter
Valid_CCSDS_Parms	CADU_Check_Tol	Number(1)	0-3 (Not NULL)	Check tolerance parameter
Valid_CCSDS_Parms	CADU_Flywheel_Tol	Number(1)	0-3 (Not NULL)	Flywheel tolerance parameter
Valid_CCSDS_Parms	CADU_Sync_Mark_ Check_Error_Tol	Number(1)	0-3 (Not NULL)	Check error tolerance parameter
Valid_CCSDS_Parms	CADU_Sync_Lock_ Error_Tol	Number(1)	0-3 (Not NULL)	Lock error tolerance parameter
Valid_CCSDS_Parms	CADU_Bit_Slip_Corr_ Extent	Number(1)	0-3 (Not NULL)	Bit slip correction extent parameter

3.2.3.2 IPC Mechanism

The IPC mechanism for the RDPS Quality and Accounting Information interface is the LPS database. The database table name consisting of the quality and accounting information is called "RDP_Acct". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

3.2.3.3 Data Transfer

The RDPS and MACS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Prior to completion of Level 0R processing, the RDPS must create a new database table record in RDP_Acct for each processed contact period. Within the new record, the following must be initialized.

Set:

- RDP_Acct(Contact_Sequence_Id, File_Version_Number, CCSDS_Parms_Id, Inverted_CADU_Cnt, Polarity_Change_Cnt, CADU_Bit_Slip_Cnt, CADU_Sync_Err_Cnt, CADU_Rcv_Cnt, CADU_Flywheel_Cnt, Fill_CADU_Cnt (**TBR**), CADU_CRC_Err_Cnt, VCDU_Header1_Corr_Err_Cnt, VCDU_Header2_Corr_Err_Cnt, VCDU_Header_Uncorr_Err_Cnt, BCH_Data_Corrected_Err_Cnt, BCH_Data_Uncorrected_Err_Cnt, BCH_Data_Corrected_Bits_Cnt (**TBR**), BCH_Ptr_Corrected_Err_Cnt, BCH_Ptr_Uncorrected_Err_Cnt, BCH_Ptr_Corrected_Bits_Cnt (**TBR**), CADU_CRC_Err_Cnt)

Where:

- RDP_Acct(Contact_Sequence_Id) = Contact sequence ID as described in Section 2.1.4, and
- RDP_Acct(File_Version_Number) = File version number as described in Section 2.1.4, and
- RDP_Acct(CCSDS_Parms_Id) = Valid_CCSDS_Parms.CCSDS_Parms_Id, and
- RDP_Acct(Inverted_CADU_Cnt) = the total count of CADUs received with inverted polarity, and
- RDP_Acct(Polarity_Change_Cnt) = the total number of polarity changes in the CADUs, and
- RDP_Acct(CADU_Bit_Slip_Cnt) = the total number of CADUs containing bit slips, and
- RDP_Acct(CADU_Sync_Err_Cnt) = the total number of CADUs containing sync errors, and
- RDP_Acct(CADU_Rcv_Cnt) = the total number of received CADUs, and
- RDP_Acct(CADU_Flywheel_Cnt) = the total number of flywheel CADUs, and
- RDP_Acct(Fill_CADU_Cnt) = the total number of fill CADUs (**TBR**), and
- RDP_Acct(VCDU_Header1_Corr_Err_Cnt) = the total number of virtual channel 1 VCDUs with correctable VCDU headers, and
- RDP_Acct(VCDU_Header2_Corr_Err_Cnt) = the total number of virtual channel 2 VCDUs with correctable VCDU headers, and

- RDP_Acct(VCDU_Header_Uncorr_Err_Cnt) = the total number of uncorrectable VCDU headers, and
- RDP_Acct(BCH_Data_Corrected_Err_Cnt) = the total number of CADUs with corrected BCH errors in the mission data zone, and
- RDP_Acct(BCH_Data_Uncorrected_Err_Cnt) = the total number of CADUs with uncorrected BCH errors in the mission data zone, and
- RDP_Acct(BCH_Data_Corrected_Bits_Cnt) = the total number of bits BCH corrected in the mission data zone (**TBR**), and
- RDP_Acct(BCH_Ptr_Corrected_Err_Cnt) = the total number of CADUs with corrected BCH errors in the data pointer area, and
- RDP_Acct(BCH_Ptr_Uncorrected_Err_Cnt) = the total number of CADUs with uncorrected BCH errors in the data pointer area, and
- RDP_Acct(BCH_Ptr_Corrected_Bits_Cnt) = the total number of bits BCH corrected in the data pointer area (**TBR**), and
- RDP_Acct(CADU_CRC_Err_Cnt) = the total number of CADUs with CRC errors.

The RDPS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database.

The MACS will extract the quality and accounting information for the contact period, upon operator command, after successful Level 0R processing of the contact period is complete. MACS must provide the following information.

Obtain (Extract the contact period Q/A information):

- RDP_Acct(CCSDS_Parms_Id, Inverted_CADU_Cnt, Polarity_Change_Cnt, CADU_Bit_Slip_Cnt, CADU_Sync_Err_Cnt, CADU_Rcv_Cnt, CADU_Flywheel_Cnt, Fill_CADU_Cnt (**TBR**), CADU_CRC_Err_Cnt, VCDU_Header1_Corr_Err_Cnt, VCDU_Header2_Corr_Err_Cnt, VCDU_Header_Uncorr_Err_Cnt, BCH_Data_Corrected_Err_Cnt, BCH_Data_Uncorrected_Err_Cnt, BCH_Data_Corrected_Bits_Cnt (**TBR**), BCH_Ptr_Corrected_Err_Cnt, BCH_Ptr_Uncorrected_Err_Cnt, BCH_Ptr_Corrected_Bits_Cnt (**TBR**))
- Valid_CCSDS_Parms(Insertion_Time, CADU_Search_Tol, CADU_Check_Tol, CADU_Flywheel_Tol, CADU_Sync_Mark_Check_Error_Tol, CADU_Sync_Lock_Error_Tol, CADU_Bit_Slip_Corr_Extent)

Where:

- RDP_Acct(Contact_Sequence_Id) = Desired contact sequence ID, and
- RDP_Acct(File_Version_Number) = Desired file version number, and
- Valid_CCSDS_Parms(CCSDS_Parms_Id) = RDP_Acct.CCSDS_Parms_Id.

3.2.3.4 Frequency

The RDPS is expected to make available to the MACS the contact period quality and accounting information upon completion of Level 0R processing of the contact period.

3.2.3.5 Sizing

Table 3-18 specifies the sizing for the RDPS Quality and Accounting interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing
- A quality and accounting retention time of 30 days
- A maximum of 6 contact periods per day
- A nominal contact period reprocessing count of **TBD** per 30 days.

Table 3-18. RDPS Quality and Accounting Information Sizing

Interface Element	Number	Size (Bytes)
Valid_CCSDS_Parms Record	1	25
Valid_CCSDS_Parms Record (30 days)	**TBD**	**TBD**
RDP_Acct Record	1	122
RDP_Acct Record (30 days)	180	**21,960**

3.2.4 MFPS

3.2.4.1 Format

The MFPS quality and accounting key / search information are detailed in table 3-19.

Table 3-19. MFPS Quality and Accounting Key / Search Information

Table	Attribute	Type	Range	Description
Sub_Intv	Contact_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Contact sequence ID
Sub_Intv	File_Version_Number	Number(10)	0-9 (Not NULL)	File version number
Sub_Intv	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID
MFP_Acct	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID

The MFPS quality and accounting information are detailed in table 3-20.

Table 3-20. MFPS Quality and Accounting Information

Table	Attribute	Type	Range	Description
Sub_Intv	MF_Start_Time	SCTS	(Not NULL)	Subinterval start time
Sub_Intv	MF_Stop_Time	SCTS	(Not NULL)	Subinterval stop time
Sub_Intv	VCID (**TBR**)	Number(1)	1,2 (Not NULL)	Virtual channel ID
MFP_Acct	Mjf_Cnt	Number(8)	0-11725 (Not NULL)	Total number of ETM+ scans for the associated subinterval

Table 3-20. MFPS Quality and Accounting Information (Cont.)

Table	Attribute	Type	Range	Description
MFP_Acct	Mjf_CADU_Inverted_ Polarity_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of CADUs received with inverted polarity for the associated subinterval
MFP_Acct	Mjf_CADU_Polarity_ Change_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of CADUs that had inverted polarity for the associated subinterval
MFP_Acct	Mjf_CADU_Bit_Slip_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of CADUs containing bit slips for the associated subinterval
MFP_Acct	Mjf_CADU_Sync_Err_ Cnt	Number(8)	0-99999999 (Not NULL)	Total number of CADUs with sync errors for the associated subinterval
MFP_Acct	Mjf_CADU_Rcvd_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of non-fill CADUs received for the associated subinterval
MFP_Acct	Mjf_CADU_Fly_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of flywheel CADUS for the associated subinterval
MFP_Acct	Mjf_CADU_Missing_Cnt (**TBR**)	Number(8)	0-99999999 (Not NULL)	Total number of missing CADUS for the associated subinterval
MFP_Acct	Mjf_CADU_RS_Corr_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of correctable Reed-Solomon errors for the associated subinterval
MFP_Acct	Mjf_CADU_RS_Uncorr_ Cnt	Number(8)	0-99999999 (Not NULL)	Total number of uncorrectable Reed-Solomon errors for the associated subinterval
MFP_Acct	Mjf_CADU_BCH_Data_ Corr_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of CADUS with corrected BCH errors in the mission data zone for the associated subinterval
MFP_Acct	Mjf_CADU_BCH_Data_ Uncorr_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of CADUS with uncorrected BCH errors in the mission data zone for the associated subinterval
MFP_Acct	Mjf_CADU_BCH_Data_ Bits_Corr_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of corrected BCH bits in the mission data zone for the associated subinterval
MFP_Acct	Mjf_CADU_BCH_Ptr_ Corr_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of CADUS with corrected BCH errors in the data pointer zone for the associated subinterval
MFP_Acct	Mjf_CADU_BCH_Ptr_ Uncorr_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of CADUS with uncorrected BCH errors in the data pointer zone for the associated subinterval

Table 3-20. MFPS Quality and Accounting Information (Cont.)

Table	Attribute	Type	Range	Description
MFP_Acct	Mjf_CADU_BCH_Ptr_ Bits_Corr_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of corrected BCH bits in the data pointer zone for the associated subinterval
MFP_Acct	Mjf_CADU_CRC_Err_ Cnt	Number(8)	0-99999999 (Not NULL)	Total number of CADUs with CRC errors for the associated subinterval
MFP_Acct	Mjf_Full_Fill_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of entirely filled ETM+ scans for the associated subinterval
MFP_Acct	Mjf_Part_Fill_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of partially filled ETM+ scans for the associated subinterval
MFP_Acct	Mjf_Time_Code_Err_Cnt	Number(8)	0-99999999 (Not NULL)	Total number of imagery time code errors for the associated subinterval

3.2.4.2 IPC Mechanism

The IPC mechanism for the MFPS Quality and Accounting Information interface is the LPS database. The database table names consisting of the quality and accounting information are called "Sub_Intv", and "MFP_Acct". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

3.2.4.3 Data Transfer

The MFPS and MACS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

Prior to completion of Level 0R processing, the MFPS must create a new database table record in MFP_Acct for each subinterval in the processed contact period. The Sub_Intv database table is detailed in section 6.2.4. Within the new record, the following must be initialized.

Set:

- Sub_Intv(Contact_Sequence_Id, File_Version_Number, Sub_Intv_Sequence_Id, MF_Start_Time, MF_Stop_Time, VCID (**TBR**))
- MFP_Acct(Sub_Intv_Sequence_Id, Mjf_Cnt, Mjf_CADU_Inverted_Polarity_Cnt, Mjf_CADU_Polarity_Change_Cnt, Mjf_CADU_Bit_Slip_Cnt, Mjf_CADU_Sync_Err_Cnt, Mjf_CADU_Rcvd_Cnt, Mjf_CADU_Fly_Cnt, Mjf_CADU_Missing_Cnt (**TBR**), Mjf_CADU_RS_Corr_Cnt, Mjf_CADU_RS_Uncorr_Cnt, Mjf_CADU_BCH_Data_Corr_Cnt, Mjf_CADU_BCH_Data_Uncorr_Cnt, Mjf_CADU_BCH_Data_Bits_Corr_Cnt, Mjf_CADU_BCH_Ptr_Corr_Cnt, Mjf_CADU_BCH_Ptr_Uncorr_Cnt, Mjf_CADU_BCH_Ptr_Bits_Corr_Cnt, Mjf_CADU_CRC_Err_Cnt, Mjf_Full_Fill_Cnt, Mjf_Part_Fill_Cnt, Mjf_Time_Code_Err_Cnt)

Where:

- Sub_Intv initialized as described in section 6.2.4, and
- MFP_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id, and
- MFP_Acct(Mjf_Cnt) = total number of ETM+ scans for the associated subinterval, and
- MFP_Acct(Mjf_CADU_Inverted_Polarity_Cnt) = total number of CADUs received with inverted polarity for the associated subinterval, and
- MFP_Acct(Mjf_CADU_Polarity_Change_Cnt) = total number of CADUs that had inverted polarity for the associated subinterval, and
- MFP_Acct(Mjf_CADU_Bit_Slip_Cnt) = total number of CADUs containing bit slips for the associated subinterval, and
- MFP_Acct(Mjf_CADU_Sync_Err_Cnt) = total number of CADUs with sync errors for the associated subinterval, and
- MFP_Acct(Mjf_CADU_Rcvd_Cnt) = total number of non-fill CADUS received for the associated subinterval, and
- MFP_Acct(Mjf_CADU_Fly_Cnt) = total number of flywheel CADUS for the associated subinterval, and
- MFP_Acct(Mjf_CADU_Missing_Cnt) = total number of missing CADUS for the associated subinterval (**TBR**), and
- MFP_Acct(Mjf_CADU_RS_Corr_Cnt) = total number of correctable Reed-Solomon errors for the associated subinterval, and
- MFP_Acct(Mjf_CADU_RS_Uncorr_Cnt) = total number of uncorrectable Reed-Solomon errors for the associated subinterval, and
- MFP_Acct(Mjf_CADU_BCH_Data_Corr_Cnt) = total number of CADUS with corrected BCH errors in the mission data zone for the associated subinterval, and
- MFP_Acct(Mjf_CADU_BCH_Data_Uncorr_Cnt) = total number of CADUS with uncorrected BCH errors in the mission data zone for the associated subinterval, and
- MFP_Acct(Mjf_CADU_BCH_Data_Bits_Corr_Cnt) = total number of corrected BCH bits in the mission data zone for the associated subinterval, and
- MFP_Acct(Mjf_CADU_BCH_Ptr_Corr_Cnt) = total number of CADUS with corrected BCH errors in the data pointer zone for the associated subinterval, and
- MFP_Acct(Mjf_CADU_BCH_Ptr_Uncorr_Cnt) = total number of CADUS with uncorrected BCH errors in the data pointer zone for the associated subinterval, and
- MFP_Acct(Mjf_CADU_BCH_Ptr_Bits_Corr_Cnt) = total number of corrected BCH bits in the data pointer zone for the associated subinterval, and

- MFP_Acct(Mjf_CADU_CRC_Err_Cnt) = total number of CADUs with CRC errors for the associated subinterval, and
- MFP_Acct(Mjf_Full_Fill_Cnt) = total number of entirely filled ETM+ scans for the associated subinterval, and
- MFP_Acct(Mjf_Part_Fill_Cnt) = total number of partially filled ETM+ scans for the associated subinterval, and
- MFP_Acct(Mjf_Time_Code_Err_Cnt) = total number of imagery time code errors for the associated subinterval.

The MFPS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database.

The MACS will extract the quality and accounting information for the contact period, upon operator command, after successful Level 0R processing of the contact period is complete. MACS must provide the following information.

Obtain (Extract the subinterval information for the contact period):

- Sub_Intv(Sub_Intv_Sequence_Id, MF_Start_Time, MF_Stop_Time, VCID (**TBR**))

Where:

- Sub_Intv(Contact_Sequence_Id) = Desired contact sequence ID, and
- Sub_Intv(File_Version_Number) = Desired file version number.

For each Sub_Intv_Sequence_Id returned from Sub_Intv:

Obtain (Extract the Q/A information for the subinterval):

- MFP_Acct(Mjf_Cnt, Mjf_CADU_Inverted_Polarity_Cnt, Mjf_CADU_Polarity_Change_Cnt, Mjf_CADU_Bit_Slip_Cnt, Mjf_CADU_Sync_Err_Cnt, Mjf_CADU_Rcvd_Cnt, Mjf_CADU_Fly_Cnt, Mjf_CADU_Missing_Cnt (**TBR**), Mjf_CADU_RS_Corr_Cnt, Mjf_CADU_RS_Uncorr_Cnt, Mjf_CADU_BCH_Data_Corr_Cnt, Mjf_CADU_BCH_Data_Uncorr_Cnt, Mjf_CADU_BCH_Data_Bits_Corr_Cnt, Mjf_CADU_BCH_Ptr_Corr_Cnt, Mjf_CADU_BCH_Ptr_Uncorr_Cnt, Mjf_CADU_BCH_Ptr_Bits_Corr_Cnt, Mjf_CADU_CRC_Err_Cnt, Mjf_Full_Fill_Cnt, Mjf_Part_Fill_Cnt, Mjf_Time_Code_Err_Cnt)

Where:

- MFP_Acct(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id.

3.2.4.4 Frequency

The MFPS is expected to make available to the MACS the subinterval quality and accounting information upon completion of Level 0R processing of the contact period.

3.2.4.5 Sizing

Table 3-21 specifies the sizing for the MFPS Quality and Accounting interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing
- A quality and accounting retention time of 30 days
- A maximum of 6 contact periods per day
- A maximum of 35 subintervals per contact period

Table 3-21. MFPS Quality and Accounting Information Sizing

Interface Element	Number	Size (Bytes)
Sub_Intv Record	1	80
Sub_Intv Record (30 days)	6,300	**504,000**
MFP_Acct Record	1	106
MFP_Acct Record (30 days)	6,300	**667,800**

Section 4. RDCS and RDPS

This section describes all information transferred between the RDCS and the RDPS.

4.1 Raw Data Capture File

This section describes the Raw Data Capture File interface between the RDCS and the RDPS.

4.1.1 Description

The raw data capture file is a single disk file containing all raw telemetry for a single contact period.

4.1.2 Format

The raw data capture file is a standard UNIX binary file. The raw telemetry contained in the file may not be aligned on byte boundaries, and it may contain "noise" and/or data drop-outs.

The size of the file could exceed 32-bit file system limits, therefore, a 64-bit file system will be required to maintain these files.

4.1.3 IPC Mechanism

The IPC mechanism for the Raw Data Capture File interface will be via standard UNIX binary Input/Output.

4.1.4 Data Transfer

There is no direct data transfer for this interface.

4.1.5 Frequency

The file will be provided in its entirety after the contact period. No guarantee to the amount of available raw telemetry is made prior to the end of the contact period.

4.1.6 Sizing

The Raw Data Capture file will be a single file. The size of the file is variable, 0-8 GB inclusive.

Section 5. RDPS and MFPS

This section describes all information transferred between the RDPS and the MFPS.

5.1 Annotated CADUs

This section describes the Annotated CADU interface between the RDPS and the MFPS.

5.1.1 Description

The Annotated CADU interface contains all CADUs processed and identified by the RDPS. The RDPS appends the CADUs with quality and accounting annotations and provides them to the MFPS.

5.1.2 Format

The annotated CADU format is detailed in the following diagram.

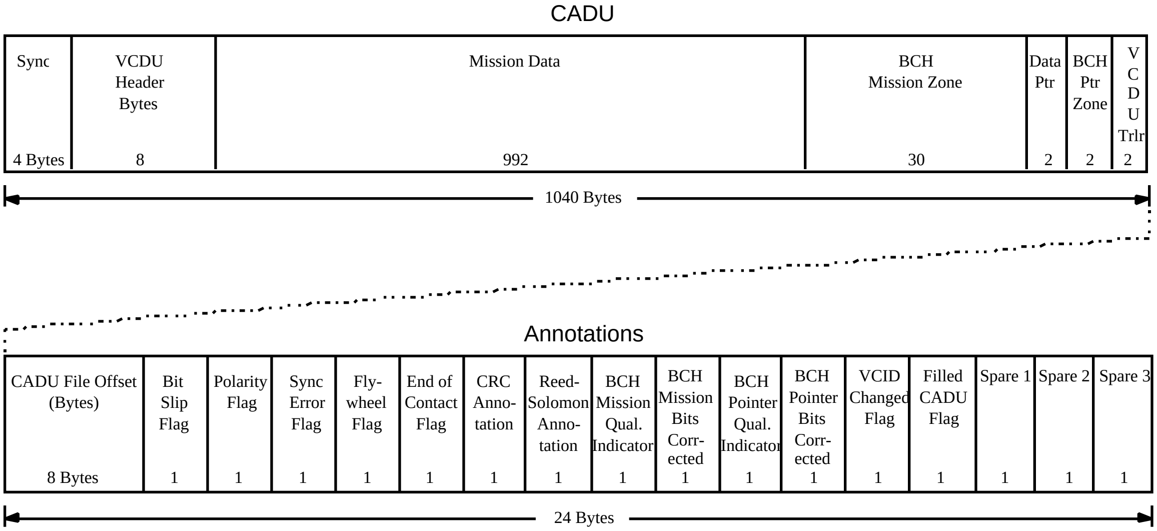


Figure 5-1. Annotated CADU

The descriptions of the annotated CADU elements are described in table 5-1.

Table 5-1. Annotated CADU Structure (lpsAnnotatedCADU)

Name	Type	Range	Description
Sync	unsigned char [4]	N/A	Synchronization pattern
VCDU_Header_Bytes	unsigned char [8]	N/A	VCDU header elements
Mission_Data	unsigned char [992]	N/A	Mission data area
BCH_Mission_Zone	unsigned char [30]	N/A	BCH mission zone associated with Mission_Data

Table 5-1. Annotated CADU Structure (lpsAnnotatedCADU) (Cont.)

Name	Type	Range	Description
Data_Pointer	unsigned char [2]	N/A	Data pointer area
BCH_Pointer_Zone	unsigned char [2]	N/A	BCH pointer zone associated with Data_Pointer
VCDU_Trailer	unsigned char [2]	N/A	CRC zone
CADU_File_Offset_In_Bytes	unsigned long long	0-ULONGLONG_MAX	Offset in bytes of this CADU from the beginning of the raw data file
Bit_Slip_Flag	unsigned char	LPS_TRUE LPS_FALSE	LPS_TRUE - Bit slip occurred LPS_FALSE - No bit slip occurred
CADU_Polarity_Flag	unsigned char	LPS_TRUE LPS_FALSE	LPS_TRUE - Original CADU was inverted LPS_FALSE - Original CADU was not inverted
CADU_Sync_Error_Flag	unsigned char	LPS_TRUE LPS_FALSE	LPS_TRUE - Bit errors in sync pattern LPS_FALSE - No bit errors in sync pattern
CADU_Flywheel_Flag	unsigned char	LPS_TRUE LPS_FALSE	LPS_TRUE - Flywheel CADU LPS_FALSE - Not a flywheel CADU
End_Of_Contact_Flag	unsigned char	LPS_TRUE LPS_FALSE	LPS_TRUE - EOF condition occurred. Last CADU to process LPS_FALSE - No EOF
CRC_Annotation	unsigned char	LPS_TRUE LPS_FALSE	LPS_TRUE - Passed CRC LPS_FALSE - Failed CRC
RS_Annotation	unsigned char	LPS_UNCORRECTABLE LPS_CORRECTED LPS_NOERRORS	Quality of CADU based on Reed-Solomon checks
BCH_Mission_Quality_Indicator	unsigned char	LPS_UNCORRECTABLE LPS_CORRECTED LPS_NOERRORS LPS_BCH_NOT_PERFORMED	Quality of mission data zone and BCH mission zone
BCH_Mission_Bits_Corrected_Count	unsigned char	0-24	Number of bit errors corrected in the mission data zone and the BCH mission data zone
BCH_Pointer_Quality_Indicator	unsigned char	LPS_UNCORRECTABLE LPS_CORRECTED LPS_NOERRORS LPS_BCH_NOT_PERFORMED	Quality of pointer data zone and BCH pointer zone
BCH_Pointer_Bits_Corrected_Count	unsigned char	0-3	Number of bit errors corrected in the pointer data zone and the BCH pointer data zone
VCID_Change_Flag	unsigned char	LPS_TRUE LPS_FALSE	LPS_TRUE - VCID changed in CADU LPS_FALSE - VCID not changed

Table 5-1. Annotated CADU Structure (lpsAnnotatedCADU) (Cont.)

Name	Type	Range	Description
Fill_CADU_Flag	unsigned char	LPS_TRUE LPS_FALSE	LPS_TRUE - Fill CADU LPS_FALSE - Not a fill CADU
Spare 1	unsigned char	N/A	N/A
Spare 2	unsigned char	N/A	N/A
Spare 3	unsigned char	N/A	N/A

Due to computer architectural restraints, the annotated CADU must start on an 8-byte memory boundary. The spare fields added to the annotations ensure that the annotated CADU size is a multiple of eight.

The structure defining the annotated CADU, and conforming to the above diagram, is located in the LPS common header file "lps_annotated_cadu.h". The structure type definition is called "lpsAnnotatedCADU".

5.1.3 IPC Mechanism

The IPC mechanism for the Annotated CADU interface is shared memory. The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs. For the purposes of this interface, it is required that the LPS common shared memory routines behave in a FIFO manner.

5.1.4 Data Transfer

The shared memory key identifier for both RDPS and MFPS is set as LPS_SHMKEY_RDPMFP, and is defined in the common header file "lps_constants.h".

The RDPS attaches to the shared memory with a call to the routine lps_ShmOpen. A call to lps_ShmGetWrBlk is then performed to obtain a shared memory block for writing. As the RDPS performs the raw data file processing, it annotates each CADU with quality and accounting information, and places it into the shared memory. When the shared memory block is full, the RDPS returns the block, in FIFO order, with a call to lps_ShmPutWrBlk to make the block available to the MFPS. The RDPS will then call lps_ShmGetWrBlk to obtain the next shared memory block for writing. Processing will continue in this manner until all of the CADUs have been processed and made available to the MFPS. The "End_Of_Contact_Flag" of the last annotated CADU within the last shared memory block will be set to LPS_TRUE to denote processing completion. If no CADUs are identified in the contact period, the RDPS will **TBD**. A call to the routine lps_ShmClose is performed to detach the RDPS from the shared memory.

The MFPS attaches to the shared memory with a call to the routine lps_ShmOpen. A call to lps_ShmGetRdBlk is then performed to obtain a shared memory block for reading. When the desired number of shared memory blocks have been obtained, the MFPS will process the annotated CADUs. When the shared memory blocks have been processed, the MFPS returns them, one at a time, with calls to lps_ShmPutRdBlk. The MFPS will then call lps_ShmGetRdBlk to obtain the next shared memory block for reading. Processing will continue in this manner until the "End_Of_Contact_Flag" is set, by the RDPS, to LPS_TRUE, and all of the annotated CADUs have been processed. A call to the routine lps_ShmClose is performed to detach the MFPS from the shared memory.

5.1.5 Frequency

The RDPS is required to provide to the MFPS a shared memory block of annotated CADUs with a maximum latency of

$$\frac{215{,}445\,\mathit{AnnCADU}}{250\mathit{s}} = \frac{\mathrm{n}\mathit{AnnCADU}}{\mathrm{T}}$$

$$\mathrm{T} = \frac{250\mathrm{n}}{215{,}445}\,\mathit{s}$$

where "n" is the number of annotated CADUs per shared memory block specified in the following subsection. This frequency is required to meet the requirements specified in the LPS SRS. The number is based on the following assumptions.

- SRS states that the RDPS will output 1 scene's worth of CADUs (215,445) every 250 seconds

5.1.6 Sizing

Table 5-2 specifies the sizing for the interface.

Table 5-2. Annotated CADU Interface Sizing

Interface Element	Number	Size (Bytes)
Annotated CADU	$\frac{200\ \text{Ann CADU}}{1\ \text{Shm Block}}$	1,064
Shared Memory Block	$\frac{15\ \text{Shm Block}}{1\ \text{Shm Seg}}$	212,800
Shared Memory Segment	1	**3,192,000**

Section 6. MFPS and PCDS

This section describes all information transferred between the MFPS and the PCDS.

6.1 PCD Information

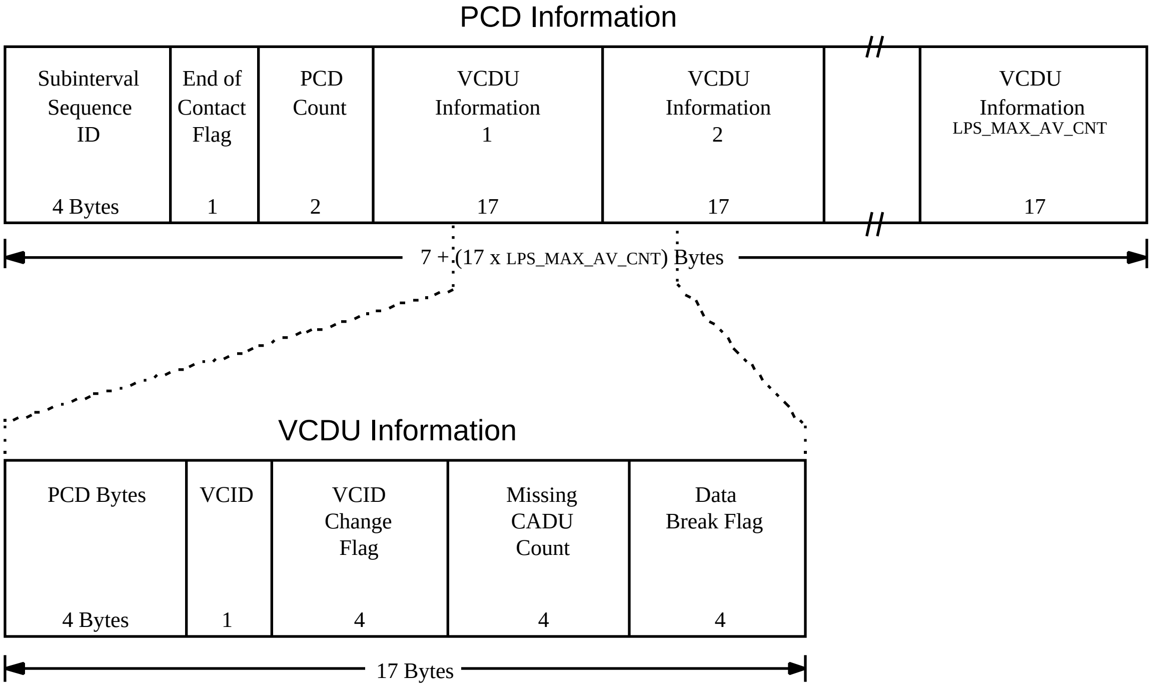
This section describes the PCD Information interface between the MFPS and the PCDS.

6.1.1 Description

The PCD Information interface contains all PCD data extracted from the raw telemetry by the MFPS. The MFPS appends the PCD bytes with VCDU accounting information annotations and provides them to the PCDS.

6.1.2 Format

The PCD information format is detailed in the following diagram.



Subinterval Sequence ID	End of Contact Flag	PCD Count	VCDU Information 1	VCDU Information 2		VCDU Information LPS_MAX_AV_CNT
4 Bytes	1	2	17	17		17

PCD Bytes	VCID	VCID Change Flag	Missing CADU Count	Data Break Flag
4 Bytes	1	4	4	4

Figure 6-1. PCD Information

The structure defining the PCD information, and conforming to the above diagram, is located in the LPS common header file "lps_pcd_info.h". The structure type definition is called "lpsPCDInfoStruct". The VCDU information structure is located in the same header file, and the type definition is called "lpsVcduInfoStruct".

The compile-time constant defining the maximum number of VCDU information structures is located in the LPS common header file "lps_constants.h". The constant is called "LPS_MAX_AV_CNT".

The descriptions of the PCD information structure, lpsPCDInfoStruct, elements are described in table 6-1.

Table 6-1. PCD Information Structure (lpsPCDInfoStruct)

Name	Type	Range	Description
lps_subIntvId	unsigned int	1-UINT_MAX	Subinterval sequence ID
lps_eocFlag	unsigned char	LPS_TRUE LPS_FALSE	LPS_TRUE - EOF condition occurred. Last PCD information structure to process LPS_FALSE - No EOF
lps_pcdCnt	unsigned short	1-USHRT_MAX	Number of VCDU information structures included in this shared memory block
lps_vcduInfo	lpsVcduInfoStruct [LPS_MAX_AV_CNT]	N/A	PCD bytes with associated VCDU accounting information

The descriptions of the VCDU information structure, lpsVcduInfoStruct, elements are described in table 6-2.

Table 6-2. VCDU Information Structure (lpsVcduInfoStruct)

Name	Type	Range	Description
lps_pcdBytes	unsigned char [4]	N/A	PCD data bytes
lps_vcid	unsigned char	1,2	The VCID of the VCDU
lps_vcduChgFlag	Boolean	LPS_TRUE LPS_FALSE	LPS_TRUE - The VCID changed on the VCDU where the PCD bytes were extracted LPS_FALSE - No VCID change
lps_missingVcduCnt	unsigned int	1-UINT_MAX	Number of CADUs missing prior to the current CADU associated with the PCD bytes
lps_dataBrk	Boolean	LPS_TRUE LPS_FALSE	LPS_TRUE - A data break occurred prior to this VCDU LPS_FALSE - No data break occurred prior to this VCDU

6.1.3 IPC Mechanism

The IPC mechanism for the PCD Information interface is shared memory. The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs. For the purposes of this interface, it is required that the LPS common shared memory routines behave in a FIFO manner.

6.1.4 Data Transfer

The shared memory key identifier for both MFPS and PCDS is set as LPS_SHMKEY_MFPPCD, and is defined in the common header file "lps_constants.h".

The MFPS attaches to the shared memory with a call to the routine lps_ShmOpen. A call to lps_ShmGetWrBlk is then performed to obtain a shared memory block for writing. As the MFPS performs the PCD data extraction, it annotates each PCD data set with accounting information, and places it into the shared memory. When the shared memory block is full, the MFPS returns the block, in FIFO order, with a call to lps_ShmPutWrBlk to make the block available to the PCDS. The MFPS will then call lps_ShmGetWrBlk to obtain the next shared memory block for writing. Processing will continue in this manner until all of the PCD data has been extracted and made available to the PCDS. The "lps_eocFlag" within the last shared memory block will be set to LPS_TRUE to denote PCD extraction completion. If no PCD data is available in the contact period, the MFPS will **TBD**. A call to the routine lps_ShmClose is performed to detach the MFPS from the shared memory.

The PCDS attaches to the shared memory with a call to the routine lps_ShmOpen. A call to lps_ShmGetRdBlk is then performed to obtain a shared memory block for reading. The PCDS processes the PCD information, and returns the shared memory read block with a call to lps_ShmPutRdBlk. The PCDS will then call lps_ShmGetRdBlk to obtain the next shared memory block for reading. Processing will continue in this manner until the "lps_eocFlag" is set, by the MFPS, to LPS_TRUE, and all of the PCD data has been processed. A call to the routine lps_ShmClose is performed to detach the PCDS from the shared memory.

6.1.5 Frequency

The MFPS is required to provide to the PCDS a shared memory block of PCD information with a maximum latency of

$$\frac{(215{,}445\,VCDUs \times (\mathrm{sizeof(lpsVCDUInfoStruct)}\,Bytes + 7\,Bytes)}{240s} = \frac{\mathrm{n}Bytes}{\mathrm{T}}$$

$$\mathrm{T} = \frac{240\mathrm{n}}{(215{,}445\,VCDUs \times (\mathrm{sizeof(lpsVCDUInfoStruct)}\,Bytes + 7\,Bytes)}\,s$$

where "n" is the size of the PCD information structure in bytes per shared memory segment specified in the following subsection. This frequency is required to meet the requirements specified in the LPS SRS. This number is based on the following assumptions:

- SRS states that the MFPS will output 1 scene's worth of PCD in 240 seconds
- 1 VCDU structure per PCD Information structure. This provides for the extra 7 bytes of overhead for each VCDU.

6.1.6 Sizing

Table 6-3 specifies the sizing for the interface. The sizes are based on the value of the compile-time constant "LPS_MAX_AV_CNT" set to 700.

Table 6-3. PCD Information Interface Sizing

Interface Element	Number	Size (Bytes)
PCD Information	$\frac{\text{1 PCD Info}}{\text{1 Shm Block}}$	11,910
Shared Memory Block	$\frac{\text{2 Shm Block}}{\text{1 Shm Seg}}$	11,910
Shared Memory Segment	1	**23,820**

6.2 Subinterval Information

This section describes the Subinterval Information interface between the MFPS and the PCDS.

6.2.1 Description

The Subinterval Information interface contains all key / search information and subinterval information determined by MFPS for the processed contact period. The MFPS provides the PCDS with notification of subinterval changes and makes available the associated subinterval information. The key / search information are not subinterval information, but information required to identify the subinterval information. The key / search information are listed in separate tables.

For the "Type" fields listed within the tables of the following subsections, an additional type has been added:

- SCTS - Spacecraft time string (Varchar2(30)) with 1/16th of a millisecond resolution, format "YYYY:DDD:HH:MM:SS.TTTTTTT" where
  - YYYY- Four digit year
  - DDD- Day of year from "001" to "366"
  - HH - Hour of day from "00" to "23"
  - MM - Minute of hour from "00" to "59"
  - SS - Second of minute from "00" to "59"
  - TTTTTTT - 1/16th of a millisecond represented in 7 fractional digits. Spacecraft clock increments are in 0.0000625 units.

6.2.2 Format

The subinterval key / search information are detailed in table 6-4.

Table 6-4. Subinterval Key / Search Information

Table	Attribute	Type	Range	Description
Sub_Intv	Contact_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Contact sequence ID
Sub_Intv	File_Version_Number	Number(10)	0-9 (Not NULL)	File version number
Sub_Intv	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID

The subinterval information is detailed in table 6-5.

Table 6-5. Subinterval Information

Table	Attribute	Type	Range	Description
Sub_Intv	Sub_Intv_Number	Number(2)	1-UCHAR_MAX (Not NULL)	The subinterval sequence number within the current contact period
Sub_Intv	MF_Start_Time	SCTS	(Not NULL)	Subinterval start time
Sub_Intv	MF_Stop_Time	SCTS	(Nullable)	Subinterval stop time
Sub_Intv	VCID	Number(1)	1,2 (Not NULL)	Virtual channel ID

6.2.3 IPC Mechanism

The IPC mechanism for the Subinterval Information interface is a combination of the LPS database and the PCD Information interface described in section 6.1. The database table name consisting of the subinterval information is called "Sub_Intv". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

6.2.4 Data Transfer

The MFPS and PCDS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

As the MFPS identifies the start of a subinterval, it creates a new database table record in Sub_Intv. Within the new record, the following must be initialized:

Set:

- Sub_Intv(Contact_Sequence_Id, File_Version_Number, Sub_Intv_Sequence_Id, Sub_Intv_Number, MF_Start_Time, VCID, MF_Stop_Time)

Where:

- Sub_Intv(Contact_Sequence_Id) = Contact sequence ID as provided in section 2.1.5, and
- Sub_Intv(File_Version_Number) = File version number as provided in section 2.1.5, and
- Sub_Intv(Sub_Intv_Sequence_Id) = A unique subinterval sequence identifier, and
- Sub_Intv(Sub_Intv_Number) = A unique subinterval sequence identifier within the contact period, and
- Sub_Intv(MF_Start_Time) = Subinterval start time, and
- Sub_Intv(VCID) = Virtual channel identifier for the current subinterval, and
- Sub_Intv(MF_Stop_Time) = NULL.

The MFPS then performs a database commit with a call to lps_db_Commit. When the information is successfully inserted into the database, the MFPS notifies the PCDS of the new subinterval information by providing the associated "Sub_Intv_Sequence_Id" through the PCD Information interface described in section 6.1. When the MFPS detects the start of a new subinterval, it updates the Sub_Intv(MF_Stop_Time) for the current subinterval information to the subinterval stop time, and begins a new subinterval information record as described above. Processing will continue in this manner until the completion of the contact period.

The PCDS will monitor the "Sub_Intv_Sequence_Id" value from the PCD Information interface described in section 6.1. When the PCDS detects that the "Sub_Intv_Sequence_Id" value has changed, it is responsible for extracting the Sub_Intv(MF_Stop_Time) from the previous subinterval, and obtains the Sub_Intv(MF_Start_Time) for the new subinterval. In order to obtain the necessary information from the LPS database table Sub_Intv, PCDS must provide the following information.

- Sub_Intv(Sub_Intv_Sequence_Id) = Subinterval sequence identifier provided by the MFPS through the PCD information interface described in section 6.1.

Processing will continue in this manner until the end of the contact period.

The routine lps_db_GetSubIntvInfo is available for extracting an entire subinterval information record from the LPS database table Sub_Intv.

6.2.5 Frequency

The MFPS is expected to provide the PCDS with the subinterval information with a maximum latency of **TBD** seconds.

6.2.6 Sizing

Table 6-6 specifies the sizing for the interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing
- A quality and accounting retention time of 30 days
- A maximum of 6 contact periods per day
- A maximum of 35 subintervals per contact period (1 scene = 1 subinterval)

Table 6-6. Subinterval Information Interface Sizing

Interface Element	Number	Size (Bytes)
Sub_Intv Record	1	82
Sub_Intv Record (30 days)	6,300	**516,600**

Section 7. MFPS and IDPS

This section describes all information transferred between the MFPS and the IDPS.

7.1 Aligned Bands

This section describes the Aligned Bands interface between the MFPS and the IDPS.

7.1.1 Description

The Aligned Bands interface contains all band detector data processed by the MFPS. The MFPS includes accounting information with the band detector data and provides them to the IDPS.

7.1.2 Format

The aligned bands format is detailed in the following diagram.

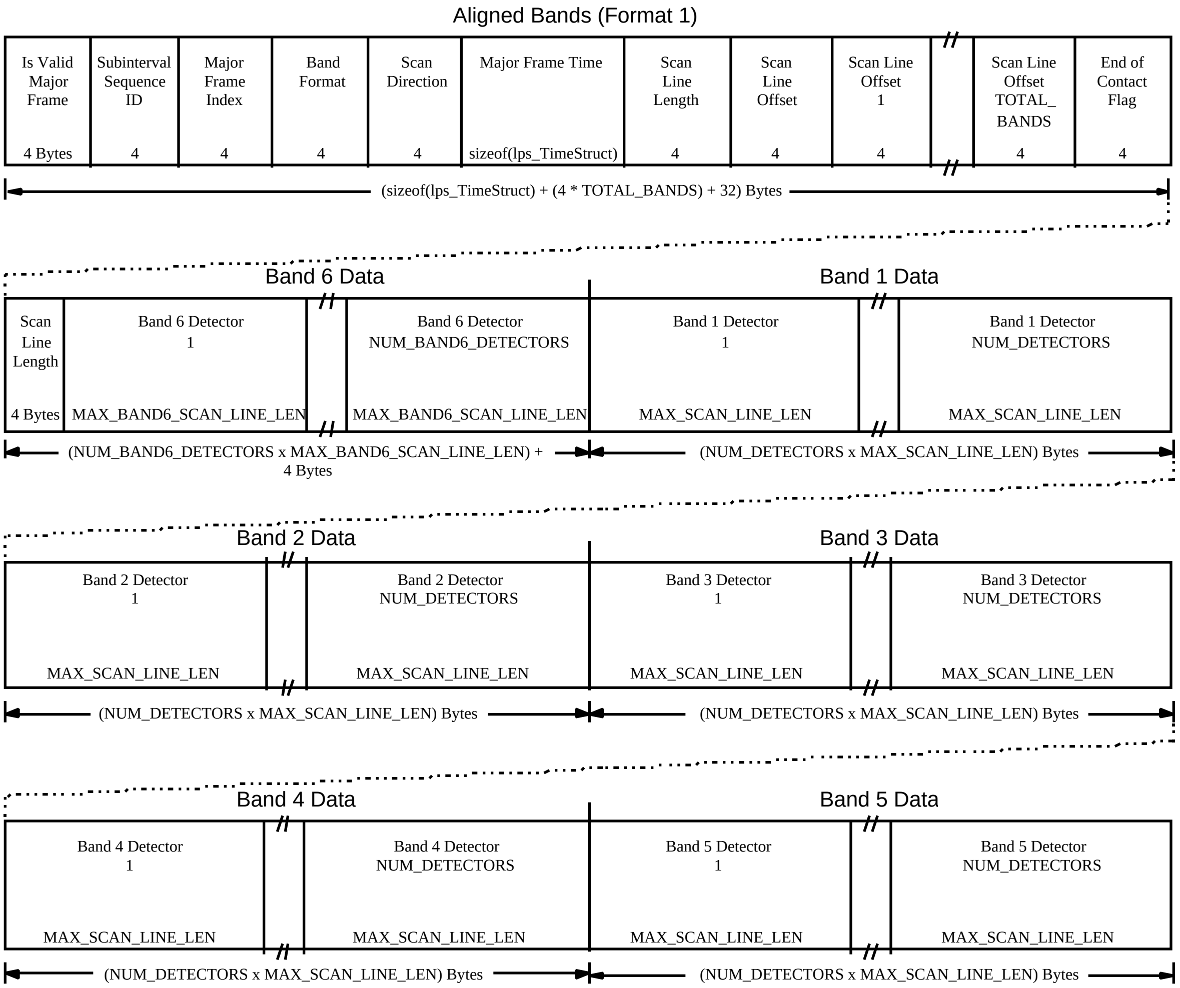


Figure 7-1. Aligned Bands (Format 1)

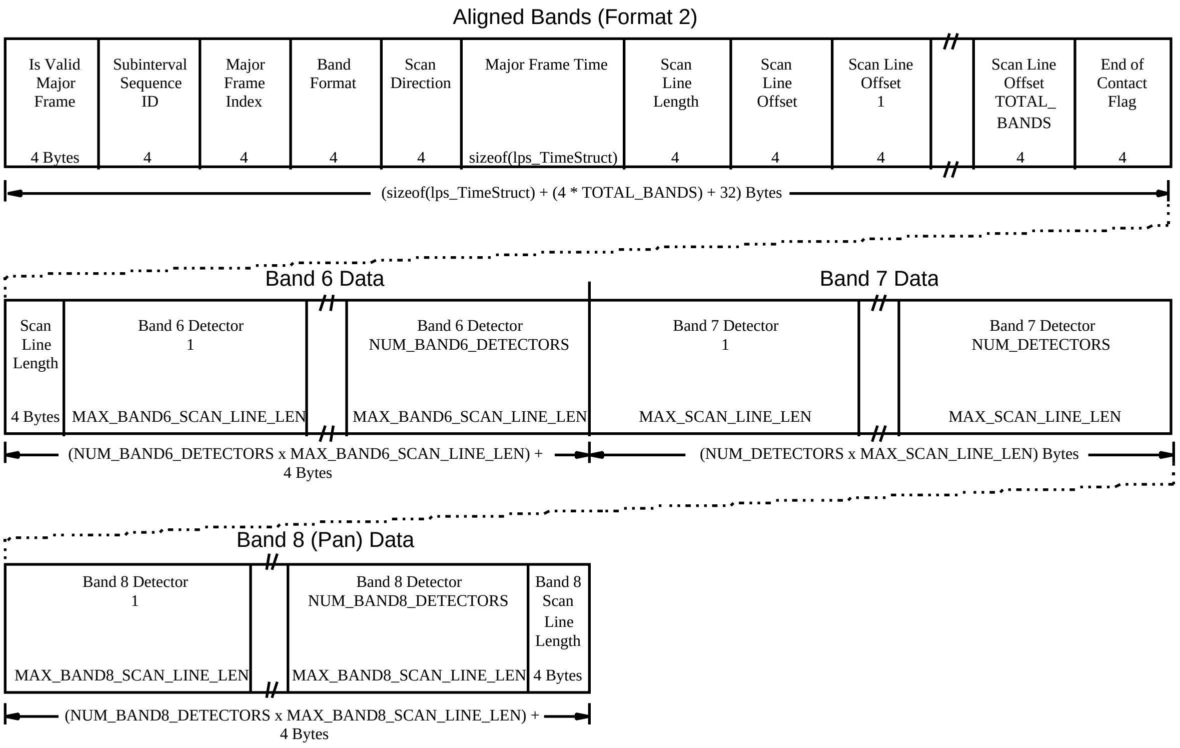


Figure 7-2. Aligned Bands (Format 2)

The descriptions of the aligned bands structure elements and supporting structure elements are described in tables 7-1 through 7-5.

Table 7-1. Major Frame Structure (lps_MajorFrame_TYPE)

Name	Type	Range	Description
Is_Valid_MF	Boolean	LPS_TRUE, LPS_FALSE	Indicates valid major frame data
Subinterval_Id	unsigned int	1-UINT_MAX	Subinterval sequence ID
MajFrameIndex	int	1-INT_MAX	The index of the major frame within the subinterval
Band_Format	lps_Band_Format_TYPE	FORMAT1, FORMAT2	Indicates the format of the band data
Scan_Direction	lps_Scan_Direction_TYPE	FORWARD, REVERSE	Indicates the direction of the scan
Major_Frame_Time	lps_TimeStruct	N/A	Major frame time
Scan_Line_Length	int	0-INT_MAX	Actual length in bytes of a scan line. Applies to all bands except for 6 and 8. Length does not include the right/left offsets.

Table 7-1. Major Frame Structure (lps_MajorFrame_TYPE) (Cont.)

Name	Type	Range	Description
Scan_Line_Offset	lps_ScanLineOffset_TYPE [TOTAL_BANDS]	N/A	Left offset of the scan data. The following indices apply: 0 = Band 1 1 = Band 2 2 = Band 3 3 = Band 4 4 = Band 5 5 = Band 6 6 = Band 7 7 = Band 8 (Pan)
End_Of_Contact_Flag	Boolean	LPS_TRUE LPS_FALSE	LPS_TRUE - EOF condition occurred. Last Aligned Bands structure to process LPS_FALSE - No EOF
Band6_Scan_Lines	lps_Band6_Scan_Data_TYPE	N/A	Band 6 detector data
Scan_Lines	lps_Format1_Scan_Data_TYPE or lps_Format2_Scan_Data_TYPE	N/A	Format specific data

Table 7-2. Band 6 Scan Data Structure (lps_Band6_Scan_Data_TYPE)

Name	Type	Range	Description
Scan_Line_Length	int	0-INT_MAX	Actual length in bytes of the band 6 scan line. Length does not include the right/left offsets.
Scan_Lines	pixel [NUM_BAND6_DETECTORS] [MAX_BAND6_SCAN_LINE_LEN]	N/A	The scan lines for each Band 6 detector.

Table 7-3. Format 1 Scan Data Structure (lps_Format1_Scan_Data_TYPE)

Name	Type	Range	Description
Scan_Lines	pixel [NUM_FORMAT1_BANDS] [NUM_DETECTORS] [MAX_SCAN_LINE_LEN]	N/A	The scan lines for each detector of each format 1 specific band. Bands are ordered from 1 to 5.

Table 7-4. Format 2 Scan Data Structure (lps_Format2_Scan_Data_TYPE)

Name	Type	Range	Description
Band7_Scan_Lines	pixel [NUM_DETECTORS] [MAX_SCAN_LINE_LEN]	N/A	The scan lines for each Band 7 detector.
Band8_Scan_Lines	pixel [NUM_BAND8_DETECTORS] [MAX_BAND8_SCAN_LINE_LEN]	N/A	The scan lines for each detector of each format 1 specific band. Bands are ordered from 1 to 5.
Band8_Scan_Line_Length	int	0-INT_MAX	Actual length in bytes of the band 8 scan line. Length does not include the right/left offsets.

Table 7-5. Time Structure (lps_TimeStruct)

Name	Type	Range	Description
Yr	short	-1, 1000-9999	Year
DayOfYr	short	1-367	Day of Year
Hr	unsigned char	0-23	Hour of Day
Min	unsigned char	0-59	Minute of Hour
Sec	unsigned char	0-60	Second of Minute (allows for leap second)
Milli	short	0-999	Millisecond of second
Sixteenth	short	0-9375	Millisecond to the 1/16th precision

The structures, enumerated types, and constants defining the aligned bands, and conforming to the above diagram and tables, are located in the LPS common header file "lps_major_frame.h". The structure type definition is called "lps_MajorFrame_TYPE". The Band 6 scan data structure is called "lps_Band6_Scan_Data_TYPE", the format 1 band scan data structure is called "lps_Format1_Scan_Data_TYPE", and the format 2 band scan data structure is called "lps_Format2_Scan_Data_TYPE". Additionally, a union type called"lps_Scan_Data_TYPE" defines the union of the format 1 and format 2 specific data structures.

The time structure defining the "Major_Frame_Time" is located in the header file "lps_Times.h", and the type definition is call "lps_TimeStruct".

7.1.3 IPC Mechanism

The IPC mechanism for the Aligned Bands interface is shared memory. The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs. For the purposes of this interface, it is required that the LPS common shared memory routines behave in a FIFO manner.

7.1.4 Data Transfer

The shared memory key identifier for both MFPS and IDPS is set as LPS_SHMKEY_MFPIDP, and is defined in the common header file "lps_constants.h".

The MFPS attaches to the shared memory with a call to the routine lps_ShmOpen. A call to lps_ShmGetWrBlk is then performed to obtain a shared memory block for writing. As the MFPS processes the band detector data, it is responsible for loading the aligned band structure, and placing it into the shared memory. When the band detector data processing is complete for each major frame, the MFPS returns the block, in FIFO order, with a call to lps_ShmPutWrBlk to make the block available to the IDPS. The MFPS will then call lps_ShmGetWrBlk to obtain the next shared memory block for writing. Processing will continue in this manner until all of the band detector data has been processed and made available to the IDPS. The "End_Of_Contact_Flag" within the last shared memory block will be set to LPS_TRUE to denote band detector data processing completion. If no band detector data is available in the contact period, the MFPS will **TBD**. A call to the routine lps_ShmClose is performed to detach the MFPS from the shared memory.

The IDPS attaches to the shared memory with a call to the routine lps_ShmOpen. A call to lps_ShmGetRdBlk is then performed to obtain a shared memory block for reading. The IDPS processes the band detector data, and returns the shared memory read block with a call to lps_ShmPutRdBlk. The IDPS will then call lps_ShmGetRdBlk to obtain the next shared memory block for reading. Processing will continue in this manner until the "End_Of_Contact_Flag" is set, by the MFPS, to LPS_TRUE, and all of the band detector data has been processed. A call to the routine lps_ShmClose is performed to detach the IDPS from the shared memory.

7.1.5 Frequency

The MFPS is required to provide to the IDPS a shared memory block of aligned bands with a maximum latency of

$$\frac{335\,MF \times \mathrm{sizeof(lps\_MajorFrame\_TYPE)}\,Bytes}{240\,s} = \frac{\mathrm{n}Bytes}{\mathrm{T}}$$

$$\mathrm{T} = \frac{240\mathrm{n}}{335\,MF \times \mathrm{sizeof(lps\_MajorFrame\_TYPE\ )}}\,s$$

where "n" is the size of the aligned band structures in bytes per shared memory block specified in the following subsection. This frequency is required to meet the requirements specified in the LPS SRS. This number is based on the following assumptions:

- SRS states that the MFPS shall output 1 scene's worth of major frames in 240 seconds
- There are 335 major frames per scene

7.1.6 Sizing

Table 7-6 specifies the sizing for the interface. The numbers presented are based on the following assumptions:

- TOTAL_BANDS = 8
- NUM_FORMAT1_BANDS = 5
- NUM_DETECTORS = 16
- MAX_SCAN_LINE_LEN = 6,600
- NUM_BAND6_DETECTORS = 8
- MAX_BAND6_SCAN_LINE_LEN = 3,300
- NUM_BAND8_DETECTORS = 32
- MAX_BAND8_SCAN_LINE_LEN = 13,200
- The lps_TimeStruct does not exceed 15 bytes
- The format 2 structure is larger than the format 1 structure by 4 Bytes

Table 7-6. Aligned Bands Interface Sizing

Interface Element	Number	Size (Bytes)
Aligned Bands	$\frac{\text{1 Aligned Bands}}{\text{1 Shm Block}}$	554,487
Shared Memory Block	$\frac{\text{2 Shm Block}}{\text{1 Shm Seg}}$	554,487
Shared Memory Segment	1	**1,108,974**

7.2 Subinterval Information

This section describes the Subinterval Information interface between the MFPS and the IDPS.

7.2.1 Description

The Subinterval Information interface contains all subinterval information determined by MFPS for the processed contact period. The MFPS provides the IDPS with notification of subinterval changes and makes available the associated subinterval information.

7.2.2 Format

See Section 6.2.2

7.2.3 IPC Mechanism

The IPC mechanism for the Subinterval Information interface is a combination of the LPS database and the Aligned Bands interface described in section 7.1. The database table name consisting of the subinterval information is called "Sub_Intv". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

7.2.4 Data Transfer

The MFPS and IDPS connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

As the MFPS identifies the start of a subinterval, it creates a new database table record in Sub_Intv. Within the new record, the following must be initialized:

- See section 6.2.4

The MFPS then performs a database commit with a call to lps_db_Commit. When the information is successfully inserted into the database, the MFPS notifies the IDPS of the new subinterval information by providing the associated "Sub_Intv_Sequence_Id" through the Aligned Bands interface described in section 7.1. When the MFPS detects the start of a new subinterval, it updates the Sub_Intv(MF_Stop_Time) for the current subinterval information to the subinterval stop time, and begins a new subinterval information record as described above. Processing will continue in this manner until the completion of the contact period.

The IDPS will monitor the "Sub_Intv_Sequence_Id" value from the Aligned Bands interface described in section 7.1. When the IDPS detects that the "Sub_Intv_Sequence_Id" value has changed, it is responsible for extracting the Sub_Intv(MF_Stop_Time) from the previous subinterval, and obtains the Sub_Intv(MF_Start_Time) for the new subinterval. In order to obtain the necessary information from the LPS database table Sub_Intv, IDPS must provide the following information.

- Sub_Intv(Sub_Intv_Sequence_Id) = Subinterval sequence identifier provided by the MFPS through the Aligned Bands interface described in section 7.1.

Processing will continue in this manner until the end of the contact period.

The routine lps_db_GetSubIntvInfo is available for extracting an entire subinterval information record from the LPS database table Sub_Intv.

7.2.5 Frequency

The MFPS is expected to provide the IDPS with the subinterval information with a maximum latency of **TBD** seconds.

7.2.6 Sizing

See Section 6.2.6

Section 8. PCDS and IDPS

This section describes all information transferred between the PCDS and the IDPS.

8.1 Scene Information

This section describes the Scene Information interface between the PCDS and the IDPS.

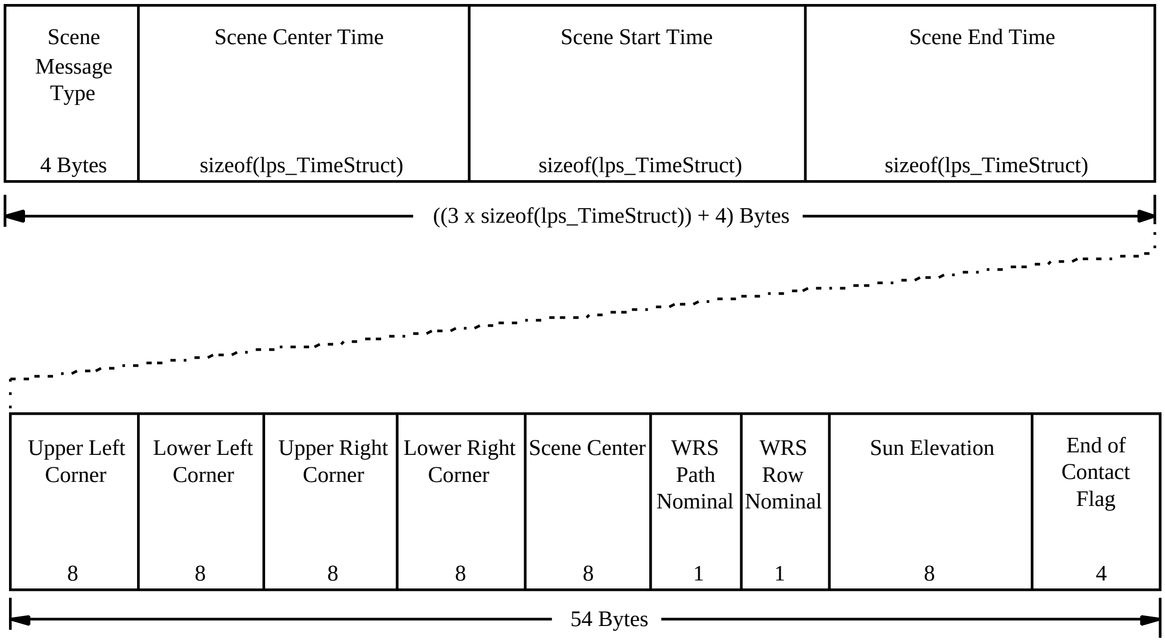
8.1.1 Description

The Scene Information interface consists of two distinct message sent from PCDS to IDPS. The "SCENE_INFO_MSG" contains the scene descriptions identified from the PCD telemetry stream by the PCDS. The "END_OF_SUBINTERVAL_MSG" indicates to the IDPS that the PCDS has identified the last scene of a subinterval. The PCDS forwards both messages to the IDPS.

8.1.2 Format

The scene information format is detailed in the following diagram.

Scene Information



Scene Message Type	Scene Center Time	Scene Start Time	Scene End Time
4 Bytes	sizeof(lps_TimeStruct)	sizeof(lps_TimeStruct)	sizeof(lps_TimeStruct)

((3 x sizeof(lps_TimeStruct)) + 4) Bytes

Upper Left Corner	Lower Left Corner	Upper Right Corner	Lower Right Corner	Scene Center	WRS Path Nominal	WRS Row Nominal	Sun Elevation	End of Contact Flag
8	8	8	8	8	1	1	8	4

54 Bytes

Figure 8-1. Scene Information

The descriptions of the scene information structure elements are described in tables 8-1 and 8-2.

Table 8-1. Scene Information Structure (lps_Scene_Info_TYPE)

Name	Type	Range	Description
scene_msg_type	lps_scene_msg_type_TYPE	SCENE_INFO_MSG, END_OF_SUBINTERVAL_MSG	Identifies the type of message
Scene_Center_Time	lps_TimeStruct	N/A	Scene center time
Scene_Start_Time	lps_TimeStruct	N/A	Time of the first scan line in the scene
Scene_End_Time	lps_TimeStruct	N/A	Time of the last scan line in the scene
Upper_Left_Corner	lps_GeoCoord_TYPE	N/A	Location of the Upper left-hand corner of the scene
Lower_Left_Corner	lps_GeoCoord_TYPE	N/A	Location of the Lower left-hand corner of the scene
Upper_Right_Corner	lps_GeoCoord_TYPE	N/A	Location of the Upper right-hand corner of the scene
Lower_Right_Corner	lps_GeoCoord_TYPE	N/A	Location of the Lower right-hand corner of the scene
Scene_Center	lps_GeoCoord_TYPE	N/A	Location of the scene center
WRS_Path_Nominal	Byte	0-233	Nominal WRS path
WRS_Row_Nominal	Byte	0-248	Nominal WRS path
Sun_Elevation	double	DBL_MIN-DBL_MAX	Solar elevation as viewed from a point on the surface of the Earth within the scene measured in degrees
End_Of_Contact_Flag	Boolean	LPS_TRUE LPS_FALSE	LPS_TRUE - EOF condition occurred. Last scene information structure to process LPS_FALSE - No EOF

Table 8-2. Geographical Coordinate Structure (lps_GeoCoord_TYPE)

Name	Type	Range	Description
latitude	float	FLT_MIN-FLT_MAX	Latitude in fractions of a degree
longitude	float	FLT_MIN-FLT_MAX	Longitude in fractions of a degree

The structures and enumerated types defining the scene information, and conforming to the above diagram and tables, are located in the LPS common header file "lps_scene_info.h". The structure type definition is called "lps_Scene_Info_TYPE". The geographical coordinate structure type defining the scene corners and the scene center is called "lps_GeoCoord_TYPE".

The time structure defining the "Scene_Center_Time", "Scene_Start_Time", and "Scene_End_Time" is located in the header file "lps_Times.h", and the type definition is called "lps_TimeStruct". The time structure detail is shown in table 7-5.

8.1.3 IPC Mechanism

The IPC mechanism for the Scene Information interface is an LPS FIFO. The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

8.1.4 Data Transfer

The FIFO key identifier for both PCDS and IDPS is set as LPS_FIFOKEY_PCDSCENEIDP, and is defined in the common header file "lps_constants.h".

The PCDS attaches to the LPS FIFO with a call to the routine lps_FIFOOpen. The PCDS provides to the IDPS two distinct messages. The first message consists of scene descriptions. The second message notifies the IDPS that the PCDS has provided the last scene description of a subinterval. The same structure type definition will be used for both messages, but the two messages are not to be combined.

As the PCDS performs scene identification, it sets the "scene_msg_type" to SCENE_INFO_MSG, sets the remaining structure elements to their appropriate values, and writes the scene information directly into the LPS FIFO with a call to lps_FIFOSend. The PCDS will continue to generate this message and call lps_FIFOSend for each identified scene. Processing will continue in this manner until all of the scenes have been identified and made available to the IDPS. The "End_Of_Contact_Flag" within the last scene information structure will be set to LPS_TRUE to denote scene identification completion.

After the PCDS has provided the last scene information message for a subinterval, it sets the "scene_msg_type" to END_OF_SUBINTERVAL_MSG, ignores the remaining structure elements, and writes the message directly into the LPS FIFO with a call to lps_FIFOSend. The PCDS will continue to generate this message and call lps_FIFOSend for each identified subinterval. Processing will continue in this manner until all of the subintervals have been identified and the messages made available to the IDPS. The last message provided to IDPS will be of the type END_OF_SUBINTERVAL_MSG. A call to the routine lps_FIFOClose is performed to detach the PCDS from the LPS FIFO.

The IDPS attaches to the LPS FIFO with a call to the routine lps_FIFOOpen. A call to lps_FIFOReceive is performed to obtain a single message from the PCDS. The IDPS will continue to call lps_FIFOReceive to obtain all of the message structures until the "End_Of_Contact_Flag" is set, by the PCDS, to LPS_TRUE, and all of the scene information has been processed. A call to the routine lps_FIFOClose is performed to detach the IDPS from the LPS FIFO.

8.1.5 Frequency

The PCDS is expected to provide each scene information structure to the IDPS with a maximum latency of 240 seconds. The PCDS is expected to provide each end-of-subinterval structure to the IDPS with a maximum latency of **TBD** seconds. These frequencies are required to meet the requirements specified in the LPS SRS.

8.1.6 Sizing

Table 8-3 specifies the sizing for the interface. The numbers presented are based on a maximum contact period of 14 minutes with at most 35 scenes contained within that contact. In addition, it is assumed the following holds true:

- The lps_TimeStruct does not exceed 15 bytes
- The PCDS writes all 35 scenes information structures into the LPS FIFO
- The PCDS writes all 35 end-of-subinterval structures into the LPS FIFO
- The IDPS has not read any scene information structures from the LPS FIFO
- The IDPS has not read any end-of-subinterval structures from the LPS FIFO

Table 8-3. Scene Information Interface Sizing

Interface Element	Number	Size (Bytes)
Scene Information	$\frac{35 \text{ Scene Info} \times 35 \text{ Subintv}}{1 \text{ Contact}}$	**126,175**

Section 9. Product File Information

This section describes all output product file information transferred between subsystems.

9.1 Product Filename Information

The Product Filename Information interface describes the output product filename information provided by the RDCS, MFPS, and MACS.

9.1.1 Description

The Product Filename Information interface consists of all key / search information and product filename information necessary for product filename generation. The RDCS, MFPS, and MACS provide the information to the MFPS, PCDS, IDPS, and MACS. The key / search information are not product filename information, but information required to identify the product filename information. The key / search information are listed in separate tables. The MFPS to MFPS and the MACS to MACS interfaces will not be described in this document.

9.1.2 RDCS

9.1.2.1 Format

The RDCS product filename key / search information are detailed in table 9-1.

Table 9-1. RDCS Product Filename Key / Search Information

Table	Attribute	Type	Range	Description
RDC_Acct	Contact_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Contact sequence ID

The RDCS product filename information is detailed in table 9-2.

Table 9-2. RDCS Product Filename Information

Table	Attribute	Type	Range	Description

RDC_Acct	Capture_Source (**TBR**)	Varchar2(2)	1I,1Q,
			2I,2Q,
			3I,3Q,
			TB,
			T1,T2,T3,T4,T5
			(Not NULL)
				Capture source ID:
				`#I - X-Band receiver #, I channel`
				`#Q - X-Band receiver #, Q channel`
				`T# - LPS string lps00#`
				`TB - LGS Bit Error Rate Tester (BERT)`

RDC_Acct	Actual_Start_Time	Date	(Not NULL)	Contact period start time

9.1.2.2 IPC Mechanism

The Inter-Process Communication (IPC) mechanism for the RDCS Product Filename Information interface is the LPS database. The database table name consisting of the product filename information is called "RDC_Acct". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

9.1.2.3 Data Transfer

The RDCS connects and disconnects from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The RDCS must attempt to disconnect from the LPS database prior to exiting execution.

As the RDCS identifies a contact period, it creates a new database table record in RDC_Acct. Within the new record, the following must be initialized.

Set:

- RDC_Acct(Contact_Sequence_Id, Capture_Source (**TBR**), Actual_Start_Time)

Where:

- RDC_Acct(Contact_Sequence_Id) = a unique contact sequence identifier, and
- RDC_Acct(Capture_Source) = capture source identifier (**TBR**), and
- RDC_Acct(Actual_Start_Time) = actual contact period start time.

The RDCS will initialize a new table record for each contact period. The RDCS must perform a database commit with a call to lps_db_Commit prior to disconnecting from the database.

Data retrieval is described in section 9.2.

9.1.2.4 Frequency

The RDCS is expected to make available the product filename information before Level 0R processing of the contact period.

9.1.2.5 Sizing

Table 9-3 specifies the sizing for the RDCS Product Filename Information interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing
- A quality and accounting retention time of 30 days
- A maximum of 6 contact periods per day

Table 9-3. RDCS Product Filename Information Sizing

Interface Element	Number	Size (Bytes)
RDC_Acct Record (1 contact)	1	15
RDC_Acct Record (30 days)	180	**2,700**

9.1.3 MFPS

9.1.3.1 Format

The MFPS product filename key / search information are detailed in table 9-4.

Table 9-4. MFPS Product Filename Key / Search Information

Table	Attribute	Type	Range	Description
Sub_Intv	Contact_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Contact sequence ID
Sub_Intv	File_Version_Number	Number(10)	0-9 (Not NULL)	File version number
Sub_Intv	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID

The MFPS product filename information are detailed in table 9-5.

Table 9-5. MFPS Product Filename Information

Table	Attribute	Type	Range	Description
Sub_Intv	Sub_Intv_Number	Number(2)	0-99 (Not NULL)	Subinterval number within the associated contact period
Sub_Intv	VCID	Number(1)	1,2 (Not NULL)	Virtual channel ID

9.1.3.2 IPC Mechanism

The IPC mechanism for the MFPS Product Filename Information interface is the LPS database. The database table name consisting of the product filename information is called "Sub_Intv". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

9.1.3.3 Data Transfer

The MFPS connects and disconnects from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The MFPS must attempt to disconnect from the LPS database prior to exiting execution.

MFPS must initialize the following as described in section 6.2.4.

Set:

- Sub_Intv(Contact_Sequence_Id, File_Version_Number, Sub_Intv_Sequence_Id, Sub_Intv_Number, VCID)

Where:

- Sub_Intv initialized as described in section 6.2.4.

9.1.3.4 Frequency

The MFPS is expected to make available the product filename information as described in section 6.2.5.

9.1.3.5 Sizing

Table 9-6 specifies the sizing for the MFPS Product Filename Information interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing
- A quality and accounting retention time of 30 days
- A maximum of 6 contact periods per day
- A maximum of 35 subintervals per contact period (1 scene = 1 subinterval)

Table 9-6. MFPS Product Filename Information Sizing

Interface Element	Number	Size (Bytes)
Sub_Intv Record	1	22
Sub_Intv Record (30 days)	6,300	**138,600**

9.1.4 MACS

9.1.4.1 Format

The MACS product filename key / search information are detailed in table 9-7.

Table 9-7. MACS Product Filename Key / Search Information

Table	Attribute	Type	Range	Description
Processing_Version_Info	Contact_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Contact sequence ID

The MACS product filename information is detailed in table 9-8.

Table 9-8. MACS Product Filename Information

Table	Attribute	Type	Range	Description
Processing_Version_ Info	File_Version_Number	Number(10)	0-9 (Not NULL)	File version number

9.1.4.2 IPC Mechanism

The Inter-Process Communication (IPC) mechanism for the MACS Product Filename Information interface is the LPS database. The database table name consisting of the product filename information is called "Processing_Version_Info". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

9.1.4.3 Data Transfer

The MACS connects and disconnects from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The MACS must attempt to disconnect from the LPS database prior to exiting execution.

The MACS will provide product filename information for the contact period, upon operator initiation of Level 0R processing. MACS must provide the following information.

Set:

- Processing_Version_Info(Contact_Sequence_Id, File_Version_Number)

Where:

- Processing_Version_Info(Contact_Sequence_Id) = RDC_Acct.Contact_Sequence_Id, and
- Processing_Version_Info(File_Version_Number) = 0 for first time processing the contact period, or other value as provided by the operator.

The MACS will initialize a new table record each time a contact period is processed. The MACS must perform a database commit, with a call to lps_db_Commit, after initializing the new record, but before starting any Level 0R subsystems.

Data retrieval is described in section 9.2.

9.1.4.4 Frequency

The MACS is expected to make available the product filename information after initializing the new record, but before starting any Level 0R subsystems for the processing of this contact period.

9.1.4.5 Sizing

Table 9-9 specifies the sizing for the MACS Product Filename Information interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing
- A quality and accounting retention time of 30 days
- A maximum of 6 contact periods per day
- A nominal contact period reprocessing count of **TBD** per 30 days

Table 9-9. MACS Product Filename Information Sizing

Interface Element	Number	Size (Bytes)
Processing_Version_Info Record	1	12
Processing_Version_Info Record (30 days)	**TBD**	**TBD**

9.2 LDTS Product File Information

The LDTS Product File Information interface describes the output product file information provided by the MFPS, PCDS, IDPS, and MACS to the LDTS.

9.2.1 Description

The LDTS Product File Information interface consists of all key / search information and product file information necessary for product file identification. The MFPS, PCDS, IDPS, and MACS provide this information to the LDTS. The key / search information are not product file information, but information required to identify the product file information. The key / search information are listed in separate tables.

9.2.2 Format

The LDTS product key / search information are detailed in table 9-10.

Table 9-10. LDTS Product File Key / Search Information

Table	Attribute	Type	Range	Description
LPS_File_Info	Sub_Intv_Sequence_Id	Number(10)	1-ULONG_MAX (Not NULL)	Subinterval sequence ID

The LDTS output product file information is detailed in table 9-11.

Table 9-11. LDTS Product File Information

Table	Attribute	Type	Range	Description
LPS_File_Info	File_Path	Varchar2(256)	(Not NULL)	The path where the output file resides
LPS_File_Info	File_Name	Varchar2(256)	(Not NULL)	The output file name
LPS_File_Info	File_Type	Varchar2(3)	(Not NULL)	The L0R file type identifier
LPS_File_Info	File_Deletion_Status	Number(1)	0-2 (Not NULL)	File deletion status: 0 - Exists 1 - Ready for Deletion 2 - Deleted

9.2.3 IPC Mechanism

The IPC mechanism for the LDTS Product File Information interface is the database table "LPS_File_Info". The LPS common routines used for this interface are mentioned in the following subsection. A detailed description of the units, along with their API can be found within the unit prologs.

9.2.4 Data Transfer

Each of the Level 0R subsystems connect and disconnect from the LPS database with calls to lps_db_Connect and lps_db_Disconnect respectively. Connection to the LPS database must be done prior to accessing the LPS database. The subsystems must attempt to disconnect from the LPS database prior to exiting execution.

The Level 0R subsystems obtain a filename and path for the specific Level 0R file with a call to lps_FileNameCreate. To register the filename and provide it to the LDTS, the responsible subsystem creates a new database table record in LPS_File_Info. Within the new record, the following must be initialized:

Set:

- LPS_File_Info(Sub_Intv_Sequence_Id, File_Path, File_Name, File_Type, File_Deletion_Status)

Where:

- LPS_File_Info(Sub_Intv_Sequence_Id) = subinterval sequence ID associated with the L0R file, and
- LPS_File_Info(File_Path) = path of the L0R file, and
- LPS_File_Info(File_Name) = L0R file name in the format L7XsssfnYYDOYHHuuv.xxx as described in table 9-12, and
- LPS_File_Info(File_Type) = value from table 9-13, and
- LPS_File_Info(File_Deletion_Status) = 0.

Table 9-12. Filename Format

Description	Remarks
L7 indicates the Landsat 7 mission	Fixed to "L7" for all files generated by LPS
X = 1, 2, or 3 for the L7 X-band data routed by LGS to an LPS String	Derived from the RDC_Acct.Capture_Source as described in section 9.1.2 (**TBR**)
sss indicates data capture ground station, for example: sss = "EDC" at Sioux Falls, SD	**TBD**

f indicates ETM+ data format:
f = 1 for Format 1 data
f = 2 for Format 2 data
	Value from Sub_Intv.VCID as described in section 9.1.3

n indicates LPS string number (1–9)	The unique number extracted from the host machine name, for example: n=1 where host name is "lps001"
YYDOYHH: indicates the Landsat 7 contact period start date and time for this file, where YY = The last two digits of the year associated with a contact period DOY = The Julian day (001 through 366) associated with the contact period HH = The hour of the contact period within a 24-hour day (00–23)	Value from RDC_Acct.Actual_Start_Time as described in section 9.1.2
uu indicates a subinterval number within this contact period (01–99)	Value from Sub_Intv.Sub_Intv_Number as described in section 9.1.3
v indicates file version number	Value from Processing_Version_Info.File_Version_Number as described in section 9.1.4
xxx indicates an LPS File type	Value from table 9-13

Table 9-13. File Identification Type

Subsystem	File	Identifier
MFPS	Calibration File	"CAL"
MFPS	Mirror Scan Correction (MSCD) File	"MSD"
PCDS	Payload Correction Data (PCD) File	"PCD"
IDPS	Band Files	"B"is i - Band ID (1-8) s - File segment number (0-4)
IDPS	Browse Files	"R"nn nn - Scene file number (01-99)
MACS	Metadata File	"MTA"

A new record is to be added for each Level 0R file within the contact period. The Level 0R subsystem is responsible for performing a database commit with a call to lps_db_Commit prior to disconnecting from the database. For ease of implementation, the routine lps_db_InsertFileInfo is available for storing the Level 0R file information into the LPS database.

The LDTS is responsible for the Level 0R file information extraction from the LPS database. All information must be obtained for a contact period, therefore, the following information must be provided.

Obtain:

- Sub_Intv(Sub_Intv_Sequence_Id)

Where:

- Sub_Intv(Contact_Sequence_Id) = contact sequence ID as provided in section 2.1.8, and
- Sub_Intv(File_Version_Number) = file version number as provided in section 2.1.8.

For each Sub_Intv_Sequence_Id returned from Sub_Intv:

Obtain:

- LPS_File_Info(File_Path, File_Name, File_Type)

Where:

- LPS_File_Info(Sub_Intv_Sequence_Id) = Sub_Intv.Sub_Intv_Sequence_Id.

9.2.5 Frequency

All file information will be provided in its entirety after the completion of the Level 0R processing by RDPS, MFPS, PCDS, and IDPS for the contact period. No guarantee to the number of available Level 0R files is made prior to the end of the contact period.

9.2.6 Sizing

Table 9-14 specifies the sizing for the interface. The numbers are based on the following:

- Oracle 7 relational DBMS data type sizing
- A quality and accounting retention time of 30 days
- A maximum of 6 contact periods per day
- A maximum of 35 subinterval files per contact period (1 scene = 1 subinterval)

Table 9-14. LDTS Product File Information Interface Sizing

Interface Element	Number	Size (Bytes)
LPS_File_Info Record	1	523
LPS_File_Info Record (1 contact)	385	**201,355**
LPS_File_Info Record (30 days)	6,300	**3,294,900**

Appendix A. Oracle 7 Data Sizing

The Oracle 7 internal data storage sizing is reflected in the following tables:

Table A-1. Oracle 7 Integer Data Type Sizing

Oracle 7 Data Type	Data Size (Bytes)
Number(1)	2
Number(2)	2
Number(3)	3
Number(4)	3
Number(5)	4
Number(6)	4
Number(8)	5
Number(10)	6

Table A-2. Oracle 7 Real Data Type Sizing

Oracle 7 Data Type	Data Size (Bytes)
Number(3,2)	3
Number(5,2)	4
Number(6,2)	4
Number(6,5)	5
Number(7,2)	5
Number(7,4)	5
Number(8,4)	5
Number(9,2)	6
Number(10,3)	6
Number(10,7)	6
Number(14,7)	9
Number(16,7)	10

Table A-3. Oracle 7 Miscellaneous Data Type Sizing

Oracle 7 Data Type	Data Size (Bytes)
Char(#)	#
Date	7
Varchar2(#)	#

Appendix B. Acronym List

ACCA	Automatic Cloud Cover Assessment
AOS	Advanced Orbiting Systems
API	Application Process Interface
BCH	Bose-Chaudhuri-Hocquenghem (EDAC)
BERT	Bit Error Rate Tester
CADU	Channel Access Data Unit
CAL	Calibration
CCA	Cloud Cover Assessment
CCB	Configuration Control Board
CCSDS	Consultative Committee for Space Data Systems
CNMOS	Consolidated Network and Mission Operations Support
CSC	Computer Sciences Corporation
DAN	Data Availability Notice
DBMS	Database Management System
DCN	Document Change Notice
DDN	Data Delivery Notice
DDS	Detailed Design Specification
DFCB	Data Format Control Book
ECS	ESDIS Core System
EDAC	Error Detection and Correction
EDC	EROS Data Center
EDC DAAC	EDC Distributed Active Archive Center
EOF	End of File
EROS	Earth Resources Observation System
ESDIS	Earth Science Data and Information System
ETM+	Enhanced Thematic Mapper Plus
F&PS	Functional and Performance Specifications
FIFO	First In First Out
GSFC	Goddard Space Flight Center
GUI	Graphical User Interface
IAS	Image Assessment System
ICD	Interface Control Document
ID	Identification
IDD	Interface Definitions Document
IDPS	Image Data Processing Subsystem
IPC	Inter-Process Communication
L0R	Level-0 Reformatted
LDTS	LPS Data Transfer Subsystem
LGS	Landsat 7 Ground Station
LMAS	Lockheed Martin Astro Space
LPS	Landsat 7 Processing System

MACS	Management and Control Subsystem
MFPS	Major Frame Processing Subsystem
MSCD	Mirror Scan Correction Data
MO&DSD	Mission Operations and Data Systems Directorate
MOC	Mission Operations Center
MOSDD	Mission Operations and Systems Development Division
MOU	Memo of Understanding
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
PCD	Payload Correction Data
PCDS	Payload Correction Data (Processing) Subsystem
RDCS	Raw Data Capture Subsystem
RDPS	Raw Data Processing Subsystem
RS	Reed-Solomon (EDAC)
SEAS	Systems, Engineering, and Analysis Support Services
SRS	Software Requirement Specification
USGS	United States Geological Survey
UTC	Universal Time Code
VCDU	Virtual Channel Data Unit
VCID	Virtual Channel ID
WRS	Worldwide Reference System